

Complete API Documentation

Generated on: 2025-10-06 12:20:19

API Documentation: API - Current Routes

Overview

Version: 4.5.2.736

Description: All endpoints that are not marked as obsolete.

Generated: 2025-10-06 12:20:19

Base URLs

- <https://dotrezapi.test.6e.navitaire.com>

API Endpoints Summary

Total Endpoints: 1024

Account (2 endpoints)

- **POST /api/nsk/v3/account/password/change** - Changes the accounts password.
- **POST /api/nsk/v1/account/password/reset** - Invokes the forgot password reset for a specific account.

Apismessage (1 endpoints)

- **POST /api/nsk/v1/apisMessage** - Takes the ApisMessage Request and sends it to core to queue a new message. Returns 201 in the response if message successfully queued and a message with Verified and Infant Counts, and 400 if errors occur (And failure text (if any) will be returned).

Apo (6 endpoints)

- **GET /api/nsk/v1/apo** - Gets the current saved ancillary pricing options data.
- **POST /api/nsk/v1/apo** - Adds new entries to the saved ancillary pricing options data.
- **PUT /api/nsk/v1/apo** - Adds new or updates existing ancillary pricing options data.
- **DELETE /api/nsk/v1/apo** - Deletes all of the saved ancillary pricing options data.
- **PUT /api/nsk/v1/apo/{inputParameterKey}** - Updates an individual saved ancillary pricing option's data.
- **DELETE /api/nsk/v1/apo/{inputParameterKey}** - Deletes a single saved ancillary pricing option's data.

Availability (6 endpoints)

- **POST /api/nsk/v3/availability/lowfare** - Searches for low fare availability with the full low fare search request. This request allows for total control over the configuration. If a property is not set, the default settings (LowFareAvailabilitySettings) are used in its place. Note low fare will not always match availability requests for certain configurations including sorting by taxes and fees.
- **GET /api/nsk/v1/availability/lowfare/estimate** - Retrieves the low fare availability estimate using a range of dates.
- **POST /api/nsk/v3/availability/lowfare/simple** - Searches for low fare availability with the simple search request. The simple search request covers the common search criteria and lets default settings (LowFareAvailabilitySettings) configure the rest. Note low fare will not always match availability requests for certain configurations included sorting by taxes and fees.
- **POST /api/nsk/v4/availability/search** - Searches for availability with the full availability search request. This request allows for total control over the configuration. If a property is not set, default values will be set instead. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opted out the filters on the availability request.
- **POST /api/nsk/v4/availability/search/simple** - Searches for availability with the simple availability search request. This request allows for easy setup for one way and round trip searches. If a property is not set, default values will be set instead just like they would in the full availability request. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opted out the filters on the availability simple request.
- **POST /api/nsk/v2/availability/search/ssr** - Searches for availability with SSR prices with the full availability search request. This request allows for total control over the configuration. If a property is not set, default values will be set instead. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out of the filters on the availability request.

Baggage (4 endpoints)

- **DELETE** /api/dcs/v1/baggage - Sets the 'Removed' status for a bag for a booking stateless.
- **POST** /api/dcs/v2/baggage - Adds baggage to a booking stateless.
- **DELETE** /api/dcs/v1/baggage/byLegKey - Sets the 'Removed' status for a bag for a booking stateless, by leg key.
- **PUT** /api/dcs/v1/baggage/checkIn - Performs an update and checks-in an individual baggage item for a specific passenger.

Bagtag (1 endpoints)

- **GET** /api/dcs/v1/bagTag/printers - Gets the bag tag printers.

Boarding (5 endpoints)

- **POST** /api/dcs/v2/boarding/{recordLocator}/legs/{legKey}/passengers/{passengerKey} - Boards a passenger on a specific leg. If the leg is NOT a part of a change of gauge flight, it is recommended to board by segment instead.
- **DELETE** /api/dcs/v1/boarding/{recordLocator}/legs/{legKey}/passengers/{passengerKey} - Unboards a passenger on a specific leg.
- **POST** /api/dcs/v2/boarding/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey} - Boards a passenger on a specific segment. A segment should not be boarded if it is a change of gauge flight.
- **DELETE** /api/dcs/v1/boarding/legs/{legKey}/passengers - Unboards all passengers on a leg.
- **POST** /api/dcs/v2/boarding/legs/{legKey}/passengers - Boards all passengers on a specific leg.

Booking (63 endpoints)

- **GET** /api/nsk/v1/booking - Get the current booking in state.
- **POST** /api/nsk/v3/booking - Commits stateful changes made and processes the booking.
- **PUT** /api/nsk/v3/booking - Commits stateful changes made and processes the booking.
- **GET** /api/nsk/v1/booking/baggageAllowances - Gets the baggage allowances for the current booking in state. Baggage usage details will only be returned if includeUsageDetails is flagged true AND the allowance behavior is NOT Default.
- **POST** /api/nsk/v1/booking/baggageAllowances/external - Gets the third-party baggage allowance calculations for the current booking in state.
- **POST** /api/nsk/v1/booking/bundle/availability - Retrieves the bundle and ssr availability for the booking in-state.
- **GET** /api/nsk/v1/booking/comments - Gets the list of comments on the in-state booking.
- **DELETE** /api/nsk/v2/booking/comments/{commentKey} - Deletes a comment from the booking.
- **POST** /api/nsk/v2/booking/divide - Divides the current booking.
- **GET** /api/nsk/v2/booking/equipment/legs/{legKey} - Retrieves equipment information for a specified leg.
- **GET** /api/nsk/v2/booking/equipment/legs/{legKey}/properties - Retrieves equipment, supported seat property and seat-dependent SSR information for a specified leg.
- **GET** /api/nsk/v2/booking/equipment/segments/{segmentKey} - Retrieves equipment information for all legs on a specified segment.
- **GET** /api/nsk/v2/booking/equipment/segments/{segmentKey}/properties - Retrieves equipment, supported seat property and seat-dependent SSR information for all legs on a specified segment.
- **POST** /api/nsk/v1/booking/eTickets - Issues new e-tickets and re-issues existing e-tickets for segments with voluntary changes.
- **PUT** /api/nsk/v1/booking/eTickets - Re-issues existing e-tickets for segments with involuntary changes such as schedule change and IROP.
- **GET** /api/nsk/v1/booking/eTickets/validation - Checks if the booking in state is valid for e-ticketing operations.
- **POST** /api/nsk/v1/booking/fareOverride/journey/{journeyKey} - Replaces an existing fare price with an amount specified.
- **GET** /api/nsk/v2/booking/fareRules - Retrieves the collection of all fare rules from the current booking in state.
- **POST** /api/nsk/v2/booking/fareRules/category50/journeys/{journeyKey}/segments/{segmentKey} - Get the category 50 fare rule information for a specific segment on a booking.
- **GET** /api/nsk/v1/booking/fareRules/fare/{fareKey} - Retrieves a fare rule from the current booking in state based on the specific fare key.
- **GET** /api/nsk/v2/booking/fareRules/journey/{journeyKey} - Retrieves the collection of all fare rules from the current booking in state for the specific journey key.
- **GET** /api/nsk/v2/booking/fareRules/segment/{segmentKey} - Retrieves the collection of all fare rules from the current booking in state for the specific segment key.
- **GET** /api/nsk/v1/booking/fee - Gets the fee price.
- **POST** /api/nsk/v1/booking/fee - Manually adds a new fee.
- **PUT** /api/nsk/v1/booking/fee/{feeKey} - Overrides a fee amount.
- **DELETE** /api/nsk/v1/booking/fee/{feeKey} - Deletes a fee.
- **PUT** /api/nsk/v1/booking/fees - Overrides multiple fee amounts.
- **GET** /api/nsk/v1/booking/fees/{feeKey}/quote - Gets the override fee quote of an existing fee.
- **DELETE** /api/nsk/v1/booking/fees/penalty - Waives the penalty fee of all journeys in the booking.
- **DELETE** /api/nsk/v1/booking/fees/spoilage - Waives the spoilage fee of all journeys in the booking.
- **DELETE** /api/nsk/v1/booking/groupName - Deletes the group name of the booking in state.
- **PUT** /api/nsk/v2/booking/groupName - Changes the group name of the booking in state.
- **GET** /api/nsk/v2/booking/hold/available - Retrieves the recommended hold date if hold is available.
- **GET** /api/nsk/v1/booking/indiaGstRegistrationRequirement - Returns true if it is expected to prompt the user for India GST registration details.
- **POST** /api/nsk/v1/booking/itinerary/print - Sends the booking itinerary notification to the specified printer. Automatically commits uncommitted changes to the state booking before queuing the booking to a print notification queue.
- **GET** /api/nsk/v1/booking/penalties - Gets the booking penalty from the booking in state.

- **DELETE** /api/nsk/v1/booking/penalties/{cancelledJourneyKey} - Waives the penalty fees of the specified cancelled journey.
- **PUT** /api/nsk/v1/booking/pointOfSale - Updates the booking point of sale.
- **PATCH** /api/nsk/v1/booking/pointOfSale - Updates the booking point of sale.
- **POST** /api/nsk/v1/booking/prepaidExternal - Creates a new prepaid external payment on the booking in state.
- **PUT** /api/nsk/v1/booking/price - Applies fares, discounts and taxes to the booking in state.
- **GET** /api/nsk/v1/booking/price/validation - Validates the fare prices on the booking in state.
- **POST** /api/nsk/v1/booking/promotion - Sets a promotion code to a booking in state.
- **PUT** /api/nsk/v1/booking/promotion - Updates a promotion code to a booking in state.
- **DELETE** /api/nsk/v1/booking/promotion - Deletes the promotion code on the booking in state.
- **POST** /api/nsk/v3/booking/queue - Adds the booking from state to a booking queue.
- **DELETE** /api/nsk/v2/booking/queue - Removes the booking in state from a booking queue.
- **DELETE** /api/nsk/v1/booking/queues/{bookingQueueKey} - Removes the booking in state from the specified booking queue item.
- **GET** /api/nsk/v1/booking/recordLocators - Gets all of the record locators on the in-state booking.
- **POST** /api/nsk/v1/booking/recordLocators - Adds a third party record locator to the booking.
- **DELETE** /api/nsk/v1/booking/recordLocators - Deletes all record locators from the booking.
- **GET** /api/nsk/v1/booking/recordLocators/{recordLocatorKey} - Gets the record locator associated with the record locator key for the in-state booking.
- **PUT** /api/nsk/v1/booking/recordLocators/{recordLocatorKey} - Replaces the data of a third party record locator on the booking.
- **PATCH** /api/nsk/v1/booking/recordLocators/{recordLocatorKey} - Patches the data of a third party record locator on the booking.
- **DELETE** /api/nsk/v1/booking/recordLocators/{recordLocatorKey} - Deletes a record locator from the booking.
- **PUT** /api/nsk/v1/booking/reinstate - Reinstates a booking with a status of hold canceled.
- **DELETE** /api/nsk/v1/booking/reset - Clears the current booking state.
- **PUT** /api/nsk/v1/booking/salesChannel - Overrides the effective sales channel for the booking in state.
- **GET** /api/nsk/v2/booking/status - Gets the status of the booking commit and returns the booking.
- **PUT** /api/nsk/v1/booking/systemCode - Updates the system code of the state booking.
- **DELETE** /api/nsk/v1/booking/systemCode - Removes the system code of the state booking.
- **GET** /api/nsk/v1/booking/validation - Evaluates the booking in state prior to commit or payment processing.
- **POST** /api/nsk/v1/booking/verifiedTravelDocuments/segments/{segmentKey}/passengers/{passengerKey} - Adds a new verified travel document to a passenger or the associated infant.

Booking/Addons (4 endpoints)

- **GET** /api/nsk/v1/booking addons/{addOnKey}/payments - Gets a specific add-on's allowed payment methods.
- **POST** /api/nsk/v1/booking addons/{addOnKey}/payments - Adds a payment to a specific add-on on the booking.
- **GET** /api/nsk/v1/booking addons/payments - Gets all the available add-ons allowed payment methods for all the add-ons on the booking in state.
- **PUT** /api/nsk/v1/booking addons sync - Syncs the prices of the Travel Commerce addons.

Booking/Boardingpasses (4 endpoints)

- **POST** /api/nsk/v3/booking/boardingpasses/journey/{journeyKey} - Retrieves the version of the boarding passes based on a specific journey.
- **POST** /api/nsk/v1/booking/boardingpasses/m2d/journey/{journeyKey} - Retrieves the specific journeys boarding passes using the M2D barcode type. This endpoint will only give a single barcode back even if the journey contains multiple segments. Note that if the journey has only one segment the M2D barcode will look just like the S2D barcode.
- **POST** /api/nsk/v1/booking/boardingpasses/s2d/journey/{journeyKey} - Retrieves the specific journeys boarding passes for each segment using the S2D barcode type. This endpoint will provide a single barcode for each segment.
- **POST** /api/nsk/v3/booking/boardingpasses/segment/{segmentKey} - Retrieves the version of the boarding passes based on a specific segment.

Booking/Checkin (8 endpoints)

- **DELETE** /api/nsk/v1/booking/checkin/journey/{journeyKey} - Removes the checkin status of passengers for a specific journey in state.
- **POST** /api/nsk/v3/booking/checkin/journey/{journeyKey} - Checks in a collection of passengers to a specific journey in state.
- **GET** /api/nsk/v2/booking/checkin/journey/{journeyKey}/requirements - Gets the checkin pre-validation requirements for a specific journey.
- **GET** /api/nsk/v1/booking/checkin/journey/{journeyKey}/status - Gets the passengers lift status for a specific journey based on the booking in state.
- **DELETE** /api/nsk/v1/booking/checkin/segment/{segmentKey} - Removes the checkin status of passengers for a specific journey's segment in state.
- **POST** /api/nsk/v3/booking/checkin/segment/{segmentKey} - Checks in a collection of passengers to a specific journey's segment in state.
- **GET** /api/nsk/v2/booking/checkin/segment/{segmentKey}/requirements - Gets the checkin pre-validation requirements for a specific journey's segment.
- **GET** /api/nsk/v1/booking/checkin/segment/{segmentKey}/status - Gets the passenger lift status for a specific segment based on the booking in state.

Booking/Contacts (16 endpoints)

- **GET** /api/nsk/v1/booking/contacts - Gets all the contacts on the booking.
- **POST** /api/nsk/v1/booking/contacts - Creates a new contact on the booking.
- **GET** /api/nsk/v1/booking/contacts/{contactTypeCode} - Gets a specific contact on the booking.
- **PUT** /api/nsk/v1/booking/contacts/{contactTypeCode} - Updates a specific contact on the booking based on type code.
- **PATCH** /api/nsk/v1/booking/contacts/{contactTypeCode} - Patches a specific contact on the booking based on type code.
- **DELETE** /api/nsk/v1/booking/contacts/{contactTypeCode} - Deletes a specific contact on the booking based on type code.
- **GET** /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers - Gets the collection of phone number for a specific contact on the booking.
- **POST** /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers - Creates a new phone number on a specific contact on the booking.
- **GET** /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType} - Gets the specific phone number from a specific contact on the booking.
- **PUT** /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType} - Updates a specific phone number on a specific contact on the booking.
- **DELETE** /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType} - Deletes a specific phone number on a specific contact on the booking.
- **GET** /api/nsk/v1/booking/contacts/primary - Gets the primary contact based on configuration.
- **POST** /api/nsk/v1/booking/contacts/primary - Creates a new primary contact based on configuration.
- **PUT** /api/nsk/v1/booking/contacts/primary - Updates the primary contact.
- **PATCH** /api/nsk/v1/booking/contacts/primary - Patches the primary contact.
- **DELETE** /api/nsk/v1/booking/contacts/primary - Delete the primary contact.

Booking/History (10 endpoints)

- **GET** /api/nsk/v1/booking/history - Gets the current booking history in state.
- **GET** /api/nsk/v1/booking/history/bagTagPrint - Gets the current bag tag print history for the booking in state.
- **POST** /api/nsk/v1/booking/history/bagTagPrint - Creates a bag tag print history for the booking in state.
- **GET** /api/nsk/v1/booking/history/flightMove - Gets the current flight move history in state.
- **GET** /api/nsk/v1/booking/history/holdDateChange - Gets the current hold date change history for the booking in state.
- **GET** /api/nsk/v1/booking/history/itinerarySent - Gets the current itinerary sent history for the booking in state.
- **GET** /api/nsk/v1/booking/history/message - Gets the current message history in state.
- **GET** /api/nsk/v1/booking/history/notification - Gets the current notification history in state.
- **GET** /api/nsk/v1/booking/history/seatAssignment - Gets the current seat assignment history in state.
- **GET** /api/nsk/v1/booking/history/segmentChange - Gets the current segment change history in state.

Booking/Journeys (23 endpoints)

- **GET** /api/nsk/v1/booking/journeys - Gets the collection of journeys on the booking in state.
- **DELETE** /api/nsk/v1/booking/journeys - Cancels all the journeys on the current booking in state.
- **GET** /api/nsk/v1/booking/journeys/{journeyKey} - Gets a single journey by the key.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey} - Cancels the specific journey from the booking in state.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey}/bundles - Removes the add-on service bundles of the requested passengers in the given journey.
- **POST** /api/nsk/v2/booking/journeys/{journeyKey}/bundles - Sells a service bundle for one or more passengers on a journey.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey}/fees/penalty - Waives the penalty fee of the specified journey.
- **GET** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/address/requirements - Gets the passenger or associated infant's destination address requirements for a specific journey for the booking in-state.
- **GET** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage - Gets all baggage for a passenger on a specific journey.
- **POST** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage - Adds baggage to the booking.
- **GET** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey} - Gets a single bag from a passenger on a journey.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey} - Removes baggage from the booking.
- **PUT** /api/nsk/v2/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey} - Updates the baggage for a passenger on the booking in state.
- **PUT** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey}/checkin - Checks in the baggage for a passenger on the booking in state.
- **PUT** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/group - Moves passenger to a different baggage group.
- **POST** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/manual - Adds manually tagged baggage to the booking.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/fees/amount - Waives fees for a specific journey and passenger, based upon the given fee type. This is an agent-only endpoint.

- **POST** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/infant - Creates an infant on a passenger in the specified journey.
- **DELETE** /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/infant - Deletes an infant from a passenger in the specified journey.
- **PATCH** /api/nsk/v1/booking/journeys/{journeyKey}/segments/{segmentKey}/status - Modifies the status of the specified segment in the journey.
- **GET** /api/nsk/v1/booking/journeys/{journeyKey}/travelDocument/requirements - Gets the travel document requirements for a specific journey for the booking in-state.
- **GET** /api/nsk/v1/booking/journeys/passengers/{passengerKey}/address/requirements - Gets the passenger or associated infant's destination address requirements for all journeys for the booking in-state.
- **GET** /api/nsk/v1/booking/journeys/travelDocument/requirements - Gets the travel document requirements for all journeys for the booking in-state.

Booking/Passengers (67 endpoints)

- **GET** /api/nsk/v1/booking/passengers - Gets the collection of passengers on the booking.
- **GET** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications - Gets passenger travel notifications for a specific passenger on the booking in state.
- **POST** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications - Creates a passenger travel notification for a specific passenger on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey} - Gets a passenger travel notification for a specific passenger on the booking in state.
- **PUT** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey} - Updates a passenger travel notification for a specific passenger on the booking in state.
- **PATCH** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey} - Patches a passenger travel notification for a specific passenger on the booking in state.
- **DELETE** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey} - Deletes a passenger travel notification for a specific passenger on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/events - Gets notification events for a specific passenger travel notification on the booking in state.
- **POST** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/events - Adds a notification event to a specific passenger travel notification on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/events/{eventType} - Get a notification event for a specific passenger travel notification on the booking in state.
- **DELETE** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/events/{eventType} - Deletes a notification event from a specific passenger travel notification on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/timedEvents - Gets notification timed events for a specific passenger travel notification on the booking in state.
- **POST** /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/timedEvents - Adds a notification timed event to a specific passenger travel notification on the booking in state.
- **GET**
/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}
- Get a notification timed event for a specific passenger travel notification on the booking in state.
- **PUT**
/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}
- Updates a notification timed event for a specific passenger travel notification on the booking in state.
- **DELETE**
/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}
- Deletes a notification timed event from a specific passenger travel notification on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey} - Gets a specific passenger on the booking.
- **PUT** /api/nsk/v3/booking/passengers/{passengerKey} - Updates a specific passenger on the booking.
- **PATCH** /api/nsk/v3/booking/passengers/{passengerKey} - Patches a specific passenger on the booking.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/addresses - Gets the collection of addresses from a specific passenger.
- **POST** /api/nsk/v2/booking/passengers/{passengerKey}/addresses - Creates a new address for a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/addresses/{addressKey} - Gets a specific address from a specific passenger.
- **PUT** /api/nsk/v3/booking/passengers/{passengerKey}/addresses/{addressKey} - Updates a specific address on a specific passenger.
- **PATCH** /api/nsk/v3/booking/passengers/{passengerKey}/addresses/{addressKey} - Patches a specific address on a specific passenger.
- **DELETE** /api/nsk/v2/booking/passengers/{passengerKey}/addresses/{addressKey} - Deletes a specific address on a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/baggage - Gets all the baggage for a specified passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey} - Retrieve a single bag from a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events - Retrieves events for a specific passenger baggage.
- **POST** /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events - Creates event information for a specific passenger baggage. A successful call immediately creates the event and does not require the booking to be committed.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events/{baggageEventKey} - Retrieves information for a specific passenger baggage event.

- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/breakdown - Gets a specific passenger price breakdown by passenger key.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/documents - Gets a specific passenger's collection of travel documents.
- **POST** /api/nsk/v2/booking/passengers/{passengerKey}/documents - Create a new travel document for a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/documents/{travelDocumentKey} - Gets a specific passenger's specific travel document.
- **PUT** /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey} - Updates a specific passenger's travel document.
- **PATCH** /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey} - Patches a specific passenger's travel document.
- **DELETE** /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey} - Deletes a specific passenger's specific travel document.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/documents/check - Gets the document check status for a specific passenger if there are any.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/emdCoupons - Gets the collection of passenger EMD coupons for a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/emdCoupons/{passengerEmdCouponKey} - Gets a specific passenger EMD coupon for a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/fees - Gets all fees for a specific passenger, based upon the given passenger key.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/fees/{passengerFeeKey} - Gets a specific fee for a specific passenger, based upon the given keys.
- **DELETE** /api/nsk/v1/booking/passengers/{passengerKey}/fees/nameChange - Waives the name change fee of the specified passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/fees/quote - Gets the override fee quote of a new fee.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/infant - Gets a specific passenger's infant.
- **POST** /api/nsk/v3/booking/passengers/{passengerKey}/infant - Creates an infant on a specific passenger.
- **PUT** /api/nsk/v3/booking/passengers/{passengerKey}/infant - Updates an infant on a specific passenger.
- **PATCH** /api/nsk/v2/booking/passengers/{passengerKey}/infant - Patches an infant on a specific passenger.
- **DELETE** /api/nsk/v2/booking/passengers/{passengerKey}/infant - Deletes an infant from a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents - Gets a specific passenger infant's collection of travel documents.
- **POST** /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents - Create a new travel document for a specific passenger infant.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey} - Gets a specific passenger infant's specific travel document.
- **PUT** /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey} - Updates a specific passenger infant travel document.
- **PATCH** /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey} - Patches a specific passenger infant travel document.
- **DELETE** /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey} - Deletes a specific passenger infant specific travel document.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents/check - Gets the document check status for a specific passenger's infant if there are any.
- **PATCH** /api/nsk/v1/booking/passengers/{passengerKey}/passengerTypeCode - Patches a specific passenger to the given passenger type.
- **GET** /api/nsk/v2/booking/passengers/{passengerKey}/seats - Gets all seats associated with a passenger.
- **DELETE** /api/nsk/v1/booking/passengers/{passengerKey}/seats/{unitKey} - Deletes a seat assignment from a specific passenger on the booking in state.
- **GET** /api/nsk/v2/booking/passengers/{passengerKey}/seats/{unitKey} - Gets passenger seat by key.
- **POST** /api/nsk/v2/booking/passengers/{passengerKey}/seats/{unitKey} - Adds a seat assignment to a specific passenger on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/seats/{unitKey}/quote - Retrieves a seat fee quote for a specific passenger.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/ssrs/{ssrCode}/price - Gets an SSR price quote on the booking in state.
- **GET** /api/nsk/v1/booking/passengers/{passengerKey}/voucher - Gets information regarding a specific voucher code and passenger. Returns what a voucher can pay for a given passenger.
- **GET** /api/nsk/v1/booking/passengers/breakdown - Gets all the passenger price breakdowns.
- **GET** /api/nsk/v1/booking/passengers/breakdown/byType - Gets all the passenger price breakdown by passenger type.
- **PUT** /api/nsk/v1/booking/passengers/groupBooking - Updates the collection of un-named or TBA passengers on the booking.

Booking/Payments (41 endpoints)

- **GET** /api/nsk/v1/booking/payments - Retrieves the booking payments on the booking in state.
- **POST** /api/nsk/v5/booking/payments - Creates a new external payment or prepaid payment on the booking in state.
- **GET** /api/nsk/v1/booking/payments/{paymentKey} - Gets a specific payment.
- **DELETE** /api/nsk/v1/booking/payments/{paymentKey} - Deletes the payment.
- **PATCH** /api/nsk/v1/booking/payments/{paymentKey}/authorize - Manually authorize a committed payment.
- **PATCH** /api/nsk/v1/booking/payments/{paymentKey}/decline - Manually decline a committed payment.
- **POST** /api/nsk/v1/booking/payments/{paymentMethod}/dcc - Gets the available direct currency codes offer. This is for making an inline DCC payment.
- **GET** /api/nsk/v1/booking/payments/allocations - Gets the payment allocations for the booking in state. Payment allocations are a collection of the payments assigned to each passenger on the booking.
- **GET** /api/nsk/v2/booking/payments/available - Retrieves the booking payment methods available for the booking in state.
- **POST** /api/nsk/v1/booking/payments/blacklist - Adds the booking payment to the blacklist.

- **PATCH** /api/nsk/v1/booking/payments/blacklist/{paymentKey} - Modifies the specified blacklisted payment.
- **DELETE** /api/nsk/v1/booking/payments/blacklist/{paymentKey} - Removes the specified payment from the blacklist.
- **GET** /api/nsk/v2/booking/payments/blacklist/{paymentKey} - Retrieves the specified blacklisted payment.
- **GET** /api/nsk/v2/booking/payments/bookingCredit - Gets credit available from a past booking.
- **POST** /api/nsk/v3/booking/payments/bookingCredit - Applies credit from a past booking.
- **POST** /api/nsk/v3/booking/payments/credit - Applies credit by reference number and type.
- **GET** /api/nsk/v2/booking/payments/credit - Gets credit available by reference number and type.
- **POST** /api/nsk/v3/booking/payments/customerCredit - Applies credit from the logged in customer to the booking in state.
- **GET** /api/nsk/v2/booking/payments/customerCredit - Gets credit available for the logged in user on the booking in state.
- **POST** /api/nsk/v6/booking/payments/dcc/{dccKey} - Creates a new external payment with direct currency conversion. This endpoint is used for inline DCC payments only.
- **POST** /api/nsk/v1/booking/payments/dcc/{dccPaymentTransactionKey} - Adds a dcc payment to the booking in state.
- **POST** /api/nsk/v1/booking/payments/fees/{feeCode} - Retrieves the payment fee for the fee code by providing the account number and payment method code for a manual form of payment.
- **GET** /api/nsk/v1/booking/payments/fees/{feeCode}/storedPayment/{storedPaymentKey} - Retrieves the payment fee for the fee code by providing the stored payment key for a stored payment. The stored payment key will be able to provide to retrieve the account number, payment method code and account number id.
- **DELETE** /api/nsk/v1/booking/payments/inProcess - Cancels an in-process payment, if one exists, from the session state. This can be used when in the middle of a 3DS, MCC, or DCC payment to remove the payment.
- **GET** /api/nsk/v3/booking/payments/mcc - Gets the available multi-currency codes available.
- **POST** /api/nsk/v6/booking/payments/mcc/{currencyCode} - Creates a new MCC payment.
- **POST** /api/nsk/v6/booking/payments/mcc/{currencyCode}/storedPayment/{storedPaymentKey} - Creates a new MCC payment from a stored payment.
- **POST** /api/nsk/v3/booking/payments/organizationCredit - Applies credit to the logged in user's organization. NOTE: It is not recommended to use a custom agency credit payment as it will not have the same behavior nor be validated like the pre-configured AG credit payment.
- **GET** /api/nsk/v2/booking/payments/organizationCredit - Gets credit available for the logged in users organization.
- **GET** /api/nsk/v1/booking/payments/refunds - Retrieves the booking payment methods available for a refund on the booking in state.
- **POST** /api/nsk/v5/booking/payments/refunds - Creates a refund for the booking in state. No longer handles Credit Shell or CS refunds.
- **POST** /api/nsk/v1/booking/payments/refunds/creditShell - Creates a credit shell refund for the booking in state.
- **POST** /api/nsk/v3/booking/payments/refunds/customerCredit - Creates a customer credit for a specified customer.
- **POST** /api/nsk/v2/booking/payments/refunds/organizationCredit - Creates an organization account refund for a payment from the booking in state.
- **POST** /api/nsk/v2/booking/payments/reversals - Reverses a payment for the booking in state.
- **POST** /api/nsk/v7/booking/payments/storedPayment/{storedPaymentKey} - Creates a payment using the stored payment information for the booking in state.
- **POST** /api/nsk/v6/booking/payments/threeDSecure - Creates a new ThreeDSecure external payment or prepaid payment on the booking in state.
- **POST** /api/nsk/v4/booking/payments/voucher - Creates a new voucher payment on the booking in state.
- **GET** /api/nsk/v3/booking/payments/voucher - Gets information regarding a specific voucher code and validates whether the referenced voucher exists as a voucher payment in the booking in state.
- **DELETE** /api/nsk/v2/booking/payments/voucher/{voucherPaymentReference} - Reverses a voucher payment on the booking in state. This refunds the claimed amount on the voucher.
- **POST** /api/nsk/v1/booking/payments/voucher/passenger/{passengerKey} - Creates a new voucher payment for a specified passenger.

Booking/Retrieve (4 endpoints)

- **GET** /api/nsk/v2/booking/retrieve - Retrieves the booking from the New Skies server based on the retrieve booking request parameters and any configured booking retrieve strategies. If the booking exists and the search criteria is allowed by one of the configured booking retrieve strategies, the booking will be loaded into the web session and returned.
 - **GET** /api/nsk/v1/booking/retrieve/{bookingKey} - Gets a specific booking by key and stores it in state.
- IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
- **GET** /api/nsk/v1/booking/retrieve/byRecordLocator/{recordLocator} - Gets a specific booking by record locator and stores it in state.
 - **GET** /api/nsk/v1/booking/retrieve/byThirdPartyRecordLocator - Gets a specific booking by third party record locator and stores it in state.

Booking/Seatmaps (3 endpoints)

- **GET** /api/nsk/v3/booking/seatmaps - Gets the list of seat maps for all the journeys for the booking in state.
- **GET** /api/nsk/v4/booking/seatmaps/journey/{journeyKey} - Gets the list of seat maps for an entire journey.
- **GET** /api/nsk/v3/booking/seatmaps/segment/{segmentKey} - Gets the list of seat maps for a specific journey's segment.

Booking/Seats (5 endpoints)

- **POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}** - Auto assigns seats to all selected passengers that do not have them for every journey on the booking. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.
- **POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/journey/{journeyKey}** - Auto assigns seats to all passengers that do not have them for a specific journey. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.
- **POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/journey/{journeyKey}/quote** - Retrieves the seat fee information of the auto-assigned seats of a passenger on a given journey.
- **POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/segment/{segmentKey}** - Auto assigns seats to all passengers that do not have them for a specific journey's segment. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.
- **POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/segment/{segmentKey}/quote** - Retrieves the seat fee information of the auto-assigned seats of a passenger on a given segment.

Booking/Segments (9 endpoints)

- **DELETE /api/nsk/v1/booking/segments/{segmentKey}** - Cancels the specific segment from the booking in state.
- **DELETE /api/nsk/v1/booking/segments/{segmentKey}/classOfService** - Restores a segment to the class of service prior to an upgrade or downgrade.
- **GET /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}** - Gets a specific passenger segment from the booking in state.
- **POST /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/infant** - Creates an infant on a passenger in the specified segment.
- **DELETE /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/infant** - Deletes an infant from a passenger in the specified segment.
- **POST /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/tickets** - Creates a ticket on a passenger segment on the booking in state.
- **PUT /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/tickets** - Updates a ticket on a passenger segment on the booking in state.
- **PUT /api/nsk/v1/booking/segments/classOfService/{classModifyKey}** - Modifies the class of service for the current booking in state to effect a segment upgrade or downgrade.
- **GET /api/nsk/v1/booking/segments/classOfService/availability** - Gets class of service availability for the booking in state.

Booking/Ssrs (11 endpoints)

- **GET /api/nsk/v1/booking/ssrs** - Retrieves the current SSRs added to the booking in state.
- **POST /api/nsk/v3/booking/ssrs** - Adds SSRs to the booking in state.
- **GET /api/nsk/v1/booking/ssrs/{ssrKey}** - Retrieves the current SSR by key.
- **PUT /api/nsk/v1/booking/ssrs/{ssrKey}** - Modify the ssr note.
- **DELETE /api/nsk/v1/booking/ssrs/{ssrKey}** - Deletes by the SSR passenger key provided by metadata.
- **POST /api/nsk/v3/booking/ssrs/{ssrKey}** - Add an SSR to the booking in state.
- **POST /api/nsk/v2/booking/ssrs/availability** - Get the ssr's available to add to the booking in state.
- **POST /api/nsk/v1/booking/ssrs/bundles/resell** - Resells any cancelled SSR bundle to the provided journey on the booking.
- **DELETE /api/nsk/v1/booking/ssrs/manual** - Deletes the SSR by the request criteria.
- **POST /api/nsk/v2/booking/ssrs/manual** - Adds SSRs to the booking in state.
- **POST /api/nsk/v1/booking/ssrs/resell** - Resells any cancelled ssrs to the provided journey on the booking.

Bookings (64 endpoints)

- **GET /api/nsk/v2/bookings** - Retrieves the booking from the New Skies server based on the retrieve booking request parameters and any configured booking retrieve strategies. If the booking exists and the search criteria is allowed by one of the configured booking retrieve strategies, the booking will be returned, but not loaded into the web session.
- **GET /api/nsk/v1/bookings/{bookingKey}/history** - Gets a itinerary history stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/bagTagPrint** - Gets the current bag tag print history for a booking, stateless.
- **POST /api/nsk/v1/bookings/{bookingKey}/history/bagTagPrint** - Creates a bag tag print history for a booking, stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/flightMove** - Gets a flight move history stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/holdDateChange** - Gets the hold date change history records for a booking.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/itinerarySent** - Gets the itinerary sent history records for a booking.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/message** - Gets a booking's message history, stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/notification** - Gets the booking notification history, stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/seatAssignment** - Gets a seat assignment history stateless.
- **GET /api/nsk/v1/bookings/{bookingKey}/history/segmentChange** - Gets a segment change history stateless.
- **POST /api/nsk/v3/bookings/{bookingKey}/queue** - Adds a booking to a booking queue.
- **DELETE /api/nsk/v2/bookings/{bookingKey}/queue** - Removes a booking from a booking queue.
- **POST /api/nsk/v1/bookings/{bookingKey}/queue/history** - Gets the booking queue history for a specific booking.
- **GET /api/nsk/v1/bookings/{recordLocator}** - Retrieves a booking itinerary stateless.

- **GET** /api/nsk/v1/bookings/{recordLocator}/account - Retrieves the booking account and collections based on the record locator.
- **POST** /api/nsk/v1/bookings/{recordLocator}/account - Creates the booking account based on the record locator and data in the request.
- **POST** /api/nsk/v1/bookings/{recordLocator}/account/collection - Creates a booking account collection and transaction based on the record locator and data in the request.
- **POST** /api/nsk/v1/bookings/{recordLocator}/account/collection/{accountCollectionKey}/transactions - Creates a booking account collection transaction based on the record locator, the account collection key, and data in the request.
- **GET** /api/nsk/v2/bookings/{recordLocator}/account/collection/{accountCollectionKey}/transactions - Retrieves the booking account collection transactions based on the record locator, the account collection key, and the request data.
- **PUT** /api/nsk/v1/bookings/{recordLocator}/account/status - Updates the booking account status.
- **GET** /api/nsk/v2/bookings/{recordLocator}/account/transactions - Retrieves all of the transactions for the booking.
- **GET** /api/nsk/v1/bookings/{recordLocator}/children - Gets children record locators of a booking not in state.
- **POST** /api/nsk/v3/bookings/{recordLocator}/comments - Adds a comment to a booking not in state.
- **POST** /api/nsk/v1/bookings/{recordLocator}/email - Sends itinerary notification via email.
- **POST** /api/nsk/v1/bookings/{recordLocator}/fareOverride/journey/{journeyKey} - Replaces an existing fare price with an amount specified.
- **PUT** /api/nsk/v1/bookings/{recordLocator}/fees/{passengerFeeKey}/confirm - Confirms the passenger fee for a given booking.
- **PUT** /api/nsk/v1/bookings/{recordLocator}/journey - Moves a journey on a stateless booking to a different journey.
- **POST** /api/nsk/v2/bookings/{recordLocator}/notification - Sends itinerary notification.
- **GET** /api/dcs/v1/bookings/{recordLocator}/passengers/{passengerKey}/documents/check - Gets the document check status for a specific passenger if there are any on a specific booking.
- **GET** /api/dcs/v1/bookings/{recordLocator}/passengers/{passengerKey}/infant/documents/check - Gets the document check status for a specific passenger's infant if there are any on a specific booking.
- **POST** /api/nsk/v1/bookings/{recordLocator}/passengers/{passengerKey}/seats/{unitKey} - Agent only. Adds a seat assignment to a specific passenger on a booking.
- **DELETE** /api/nsk/v1/bookings/{recordLocator}/passengers/{passengerKey}/seats/{unitKey} - Agent only. Deletes a seat assignment from a specific passenger on a booking.
- **POST** /api/nsk/v1/bookings/{recordLocator}/queues/{queueCode} - Retrieves the details for the booking queues of a specific booking.
- **DELETE** /api/nsk/v1/bookings/checkin/{recordLocator}/journey/{journeyKey} - Removes the checkin status of passengers for a specific journey stateless.
- **POST** /api/nsk/v3/bookings/checkin/{recordLocator}/journey/{journeyKey} - Checks in a collection of passengers to a specific journey stateless.
- **GET** /api/nsk/v2/bookings/checkin/{recordLocator}/journey/{journeyKey}/requirements - Gets the checkin pre-validation requirements for a specific journey version 2.
- **GET** /api/nsk/v1/bookings/checkin/{recordLocator}/journey/{journeyKey}/status - Gets the passengers lift status stateless for a specific journey based on the record locator.
- **DELETE** /api/nsk/v1/bookings/checkin/{recordLocator}/segment/{segmentKey} - Removes the checkin status of passengers for a specific journey's segment stateless.
- **POST** /api/nsk/v3/bookings/checkin/{recordLocator}/segment/{segmentKey} - Checks in a collection of passengers to a specific journeys segment stateless.
- **GET** /api/nsk/v2/bookings/checkin/{recordLocator}/segment/{segmentKey}/requirements - Gets the checkin pre-validation requirements for a specific bookings journeys segment version 2.
- **GET** /api/nsk/v1/bookings/checkin/{recordLocator}/segment/{segmentKey}/status - Gets the passengers lift status stateless for a specific segment based on the record locator.
- **POST** /api/nsk/v1/bookings/queues - Queues the bookings by inventory leg.
- **DELETE** /api/nsk/v1/bookings/queues/{bookingQueueKey} - Removes a booking from the specified booking queue item.
- **GET** /api/nsk/v2/bookings/queues/{queueCode}/items - Retrieves the booking queue items associated with the queue code.
- **POST** /api/nsk/v2/bookings/quote - Retrieves an itinerary quote based on the request.
- **GET** /api/nsk/v2/bookings/searchByAgency - Find booking by agency.
- **GET** /api/nsk/v2/bookings/searchByAgent - Find booking by agent.
- **GET** /api/nsk/v2/bookings/searchByAgentCode - Find booking by the agent code.
- **GET** /api/nsk/v3/bookings/searchByBagTag - Find booking by a bag tag. The StartUtc needs to be within 7 days before the created date of the booking. If the StartUtc is set to DateTime.MinValue (1/1/0001) or not set at all, the value will be set to the current date, UTC.
- **GET** /api/nsk/v2/bookings/searchByContact - Find booking by contact.
- **GET** /api/nsk/v2/bookings/searchByContactCustomerNumber - Find booking by contact customer number.
- **POST** /api/nsk/v1/bookings/searchByCreditCard - Find booking by credit card number.
- **GET** /api/nsk/v2/bookings/searchByCustomerNumber - Find booking by customer number.
- **GET** /api/nsk/v2/bookings/searchByDate - Find booking by date.
- **GET** /api/nsk/v2/bookings/searchByDocument - Find booking by document.
- **GET** /api/nsk/v2/bookings/searchByEmail - Find booking by email.
- **GET** /api/nsk/v1/bookings/searchByExternalPayment - Used to search the list of bookings that have used the account from the payment. Only External Payments should be used for searching.
- **GET** /api/nsk/v2/bookings/searchByLastName - Find booking by last name.
- **GET** /api/nsk/v2/bookings/searchByPhone - Find booking by phone.
- **GET** /api/nsk/v2/bookings/searchByRecordLocator - Find booking by record locator.

- **GET /api/nsk/v2/bookings/searchByReferenceNumber** - Find booking by reference number. The reference number is used as identifier by New Skies Rail customers for purposes of ticketing and check in.
- **GET /api/nsk/v3/bookings/searchByThirdPartyRecordLocator** - Find booking by a third party record locator.
- **GET /api/nsk/v1/bookings/watchList/{bookingKey}** - Retrieves the watch list for a specific booking.

Bundles (3 endpoints)

- **GET /api/nsk/v1/bundles/applications/{bundleApplicationKey}/details** - Retrieves a specific bundle application's details.
- **GET /api/nsk/v1/bundles/rules/{bundleRuleCode}/details** - Retrieves the list of bundle applications associated with the bundle rule.
- **GET /api/nsk/v1/bundles/sets/{bundleSetCode}/details** - Retrieves the list of bundle codes associated with the bundle set.

Collection (2 endpoints)

- **PUT /api/nsk/v1/collection/{accountCollectionKey}/expirationDate** - Updates the expiration of the account collection based on the account collection key and data in the request.
- **GET /api/nsk/v2/collection/{accountCollectionKey}/transactions** - Retrieves the credit transactions based on the account collection key and the request data.

Contacts (1 endpoints)

- **GET /api/nsk/v1/contacts/searchByPhoneNumber** - Gets the latest contact in a booking with a specific phone number.

Contents (2 endpoints)

- **GET /api/nsk/v1/contents/news** - Retrieves the available news metadata.
- **POST /api/nsk/v1/contents/news** - Add a new news content item.

Currency (1 endpoints)

- **GET /api/nsk/v1/currency/converter** - Calculates currency conversions.

Equipment (2 endpoints)

- **POST /api/dcs/v1/equipment/swap** - Exchanges one equipment for another on an existing flight.
- **POST /api/dcs/v1/equipment/tailNumber** - Assigns a tail number to a specific leg.

Etickets (4 endpoints)

- **GET /api/nsk/v2/eTickets/{eTicketNumber}** - Retrieves the passenger's e-ticket and its details by providing the ticket number, hosted carrier code and the alternate carrier code from the e-ticket database if the ticket is OA validated (issued by the other airline not by the hosted carrier).
- **GET /api/nsk/v1/eTickets/{eTicketNumber}/coupons** - Retrieves all coupons associated with the e-ticket.
- **GET /api/nsk/v1/eTickets/{eTicketNumber}/coupons/{couponNumber}** - Retrieves a single coupon associated with the e-ticket.
- **PUT /api/nsk/v2/eTickets/{eTicketNumber}/coupons/{couponNumber}** - Updates a coupon contained in an e-ticket.

Farerules (2 endpoints)

- **GET /api/nsk/v1/fareRules/{fareAvailabilityKey}** - Gets the fare rules from the server based on a fare availability key.
- **POST /api/nsk/v2/fareRules/category50/journeys/{journeyKey}/segments/{segmentKey}** - Get the category 50 fare rule information after an availability call.

Graph (2 endpoints)

- **POST /api/v2/graph/{queryName}** - Invokes a graph query and returns the response.
- **POST /api/nsk/v1/graph/{queryName}** - Invokes a graph query configured in utilities.

Health (1 endpoints)

- **GET /api/v3/health** - Performs the health check.

Inventory (9 endpoints)

- **GET /api/dcs/v2/inventory/legs/{legKey}** - Gets the stateless inventory leg.
- **PATCH /api/dcs/v2/inventory/legs/{legKey}** - Patches an inventory leg's data, its nests' data, and its classes' data based on information in the request.
- **GET /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}** - Gets an inventory leg nest by key.
- **PATCH /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}** - Patches an inventory leg nest's data based on information in the request.

- **GET** /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}/classes/{legClassKey} - Gets an inventory leg nest class by key.
- **PATCH** /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}/classes/{legClassKey} - Patches an inventory leg nest class's data based on information in the request.
- **GET** /api/dcs/v1/inventory/routes/{tripKey}/classes - Gets the classes for a route, stateless.
- **PATCH** /api/dcs/v1/inventory/routes/{tripKey}/classes/{classOfService} - Patches an inventory route class's data based on information in the request.
- **DELETE** /api/dcs/v1/inventory/routes/{tripKey}/classes/{classOfService}/authorizedUnit - Deletes an inventory route class's authorized unit.

Manifest (10 endpoints)

- **GET** /api/nsk/v2/manifest - Gets the manifest trip information response list for the specified manifest trip information search request.
- **GET** /api/nsk/v1/manifest/{legKey} - Gets the manifest for the specified leg.
- **GET** /api/nsk/v1/manifest/{legKey}/bags - Gets the manifest Passenger Bags for the specified leg. Sorted by recordLocator and then by Passenger.weightType 0 = Default, 1 = Pounds, 2 = Kilograms, (2 is current default for backwards compatibility)
- **GET** /api/nsk/v1/manifest/{legKey}/bookings - Gets the manifest bookings data for the specified leg.
- **GET** /api/nsk/v1/manifest/{legKey}/governmentSecurityInfo - Gets the manifest Passenger Government Security Information for the specified leg.
- **GET** /api/nsk/v2/manifest/{legKey}/passengerDetails - Gets the passenger details for the specified leg.
- **GET** /api/dcs/v1/manifest/{legKey}/passengerDetails/weightBalance - Gets the weight balance passenger details for the specified leg.
- **GET** /api/nsk/v1/manifest/{legKey}/passengerSeatAssignments - Gets a list of seat assignment reports for the specified leg.
- **GET** /api/nsk/v1/manifest/{legKey}/segments - Gets the manifest segment data for the specified leg.
- **GET** /api/nsk/v1/manifest/{legKey}/ssrs - Gets the manifest SSRs for the specified leg.

Messages (6 endpoints)

- **POST** /api/nsk/v1/messages - Adds a new message item.
- **DELETE** /api/nsk/v1/messages - Deletes a collection of message items.
- **GET** /api/nsk/v2/messages - Retrieves a collection of teletype messages based on search criteria.
- **GET** /api/nsk/v1/messages/{messageKey} - Retrieves a specific message item.
- **DELETE** /api/nsk/v1/messages/{messageKey} - Deletes a message item.
- **POST** /api/nsk/v1/messages/teletype - Adds a new generic teletype message item.

OnetimeTravelNotifications (3 endpoints)

- **POST** /api/nsk/v2/oneTimeTravelNotifications - Creates a one time travel notification.
- **GET** /api/nsk/v2/oneTimeTravelNotifications/{subscriptionNumber} - Gets a one time travel notification through a subscription number.
- **DELETE** /api/nsk/v2/oneTimeTravelNotifications/{subscriptionNumber} - Deletes a one time travel notification through a subscription number.

Organizationgroup (1 endpoints)

- **POST** /api/nsk/v2/organizationGroup - Creates a new organization group and adds an organization to it.

Organizationgroups (4 endpoints)

- **PUT** /api/nsk/v1/organizationGroups/{organizationGroupCode} - Modifies an organization group by updating the group name or adding an organization to the organization group.
- **DELETE** /api/nsk/v1/organizationGroups/{organizationGroupCode} - Deletes an organization group. This should not be used if the organization group needs to exist.
- **PUT** /api/nsk/v1/organizationGroups/{organizationGroupCode}/organizations - Updates an organization group by removing all organizations
- **PUT** /api/nsk/v1/organizationGroups/{organizationGroupCode}/organizations/{organizationCode} - Updates an organization group by removing an organization

Organizations (2 endpoints)

- **GET** /api/nsk/v1/organizations/groups - Retrieves the organization groups.
- **GET** /api/nsk/v1/organizations/groups/{organizationGroupCode} - Retrieves the details of an organization group.

Organizations2 (35 endpoints)

- **POST** /api/nsk/v1/organizations2 - Create a new organization.
- **GET** /api/nsk/v2/organizations2 - Searches for organizations.
- **GET** /api/nsk/v1/organizations2/{organizationCode} - Retrieves the organization.
- **PUT** /api/nsk/v1/organizations2/{organizationCode} - Updates an organization.
- **PATCH** /api/nsk/v1/organizations2/{organizationCode} - Updates an organization.

- **GET** /api/nsk/v1/organizations2/{organizationCode}/account - Retrieves the organization account and collections based on the organization code.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/account - Creates an organization account based on the organization code.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/account/childAccountTransactions - Adjust the available amount of a child account.
- **PUT** /api/nsk/v1/organizations2/{organizationCode}/account/childAccountTransactions - Adjust the credit limit of a child account.
- **PUT** /api/nsk/v1/organizations2/{organizationCode}/account/status - Updates the organization account status.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/account/transactions - Creates account transactions based on data in the request and the organization code.
- **GET** /api/nsk/v2/organizations2/{organizationCode}/account/transactions - Retrieves the organization account transactions based on the organization code and data in the request.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/commissionRates - Create a commission rate.
- **GET** /api/nsk/v1/organizations2/{organizationCode}/commissionRates/{commissionRateCode} - Get a commission rate.
- **DELETE** /api/nsk/v1/organizations2/{organizationCode}/commissionRates/{commissionRateCode} - Delete a commission rate.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers - Create a company phone number.
- **PUT** /api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers/{phoneNumberType} - Updates a company phone number.
- **DELETE** /api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers/{phoneNumberType} - Deletes a company phone number.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers - Create a contact phone number.
- **PUT** /api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers/{phoneNumberType} - Updates a contact phone number.
- **DELETE** /api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers/{phoneNumberType} - Deletes a contact phone number.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/externalAccounts - Create an external account.
- **GET** /api/nsk/v1/organizations2/{organizationCode}/externalAccounts/{externalAccountKey} - Get an external account.
- **DELETE** /api/nsk/v1/organizations2/{organizationCode}/externalAccounts/{externalAccountKey} - Delete an external account.
- **GET** /api/nsk/v1/organizations2/{organizationCode}/processSchedules - Returns a list of Process Schedules for the organization.
- **POST** /api/nsk/v1/organizations2/{organizationCode}/processSchedules/{processScheduleId} - Adds a process schedule to the organization.
- **DELETE** /api/nsk/v1/organizations2/{organizationCode}/processSchedules/{processScheduleId} - Deletes a process schedule from the organization.
- **GET** /api/nsk/v1/organizations2/allotments - Retrieves all allotments for the specified leg.
- **GET** /api/nsk/v1/organizations2/allotments/{allotmentBasisCode} - Retrieves a single allotment for the specified leg using the given allotment basis code.
- **GET** /api/nsk/v1/organizations2/allotments/byAllotmentBasisCode/{allotmentBasisCode} - Retrieves all allotments for the specified allotment basis code.
- **POST** /api/nsk/v1/organizations2/contracts/basis/{allotmentBasisCode}/allotments/legs/{legKey} - Creates an allotment for the specified leg.
- **PUT** /api/nsk/v1/organizations2/contracts/basis/{allotmentBasisCode}/allotments/legs/{legKey} - Updates an allotment for the specified leg.
- **GET** /api/nsk/v2/organizations2/processSchedule/{processScheduleId} - Returns a single Process Schedule object.
- **GET** /api/nsk/v2/organizations2/processSchedules - Returns a list of active process schedules.
- **POST** /api/nsk/v1/organizations2/register - Requests to register a new organization.

Payments (1 endpoints)

- **GET** /api/nsk/v1/payments - Gets the payment data based on the specified bank identification number and date range.

Persons (94 endpoints)

- **POST** /api/nsk/v1/persons - Creates a new person record (used by agents).
- **GET** /api/nsk/v2/persons - Search person records.
- **GET** /api/nsk/v1/persons/{personKey} - Retrieves a specific person by key.
- **PUT** /api/nsk/v1/persons/{personKey} - Updates the person record basic information.
- **PATCH** /api/nsk/v1/persons/{personKey} - Patches the person records basic information.
- **DELETE** /api/nsk/v1/persons/{personKey} - Deletes a person by setting the record to terminated.
- **GET** /api/nsk/v1/persons/{personKey}/account - Retrieves the person account and credits based on the person key.
- **POST** /api/nsk/v1/persons/{personKey}/account - Creates the person account based on the person key and data in the request.
- **POST** /api/nsk/v1/persons/{personKey}/account/collection - Creates an account collection based on the person key and data in the request.
- **POST** /api/nsk/v1/persons/{personKey}/account/collection/{accountCollectionKey}/transactions - Creates an account collection transaction based on the person key, the account collection key, and data in the request.
- **GET** /api/nsk/v2/persons/{personKey}/account/collection/{accountCollectionKey}/transactions - Retrieves the person account collection transactions based on the person key, the account collection key, and data in the request.
- **PUT** /api/nsk/v1/persons/{personKey}/account/status - Updates the person account status.
- **GET** /api/nsk/v2/persons/{personKey}/account/transactions - Retrieves all persons account transactions.
- **GET** /api/nsk/v1/persons/{personKey}/addresses - Gets the collection of addresses for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/addresses - Creates a new address for a specific person.

- **GET** /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey} - Gets a specific address for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey} - Updates a specific address for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey} - Patches a specific address for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey} - Deletes a specific address for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/affiliations - Gets all affiliations for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/affiliations - Creates a new affiliation for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey} - Gets a specific affiliation for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey} - Updates a specific affiliation for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey} - Deletes a specific affiliation for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/aliases - Gets the collection of aliases for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/aliases - Creates a new alias for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey} - Gets a specific alias for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey} - Updates a specific alias for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey} - Patches a specific alias for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey} - Deletes a specific alias for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/comments - Gets the collection of comments for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/comments - Creates a new comment for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/comments/{personCommentKey} - Gets a specific comment for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/comments/{personCommentKey} - Updates a specific comment for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/comments/{personCommentKey} - Patches a specific comment for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/comments/{personCommentKey} - Deletes a specific comment for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/emails - Gets the collection of email addresses for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/emails - Creates a new email address for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey} - Gets a specific email address for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey} - Updates a specific email address for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey} - Patches a specific email address for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey} - Deletes a specific email address for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/information - Gets all information from the logged in user's person.
- **POST** /api/nsk/v1/persons/{personKey}/information - Creates a new information on the logged in user's person.
- **GET** /api/nsk/v1/persons/{personKey}/information/{personInformationKey} - Gets a specific information from the logged in user's person.
- **PUT** /api/nsk/v1/persons/{personKey}/information/{personInformationKey} - Updates a specific information on the logged in user's person.
- **PATCH** /api/nsk/v1/persons/{personKey}/information/{personInformationKey} - Patches a specific information on the logged in user's person.
- **DELETE** /api/nsk/v1/persons/{personKey}/information/{personInformationKey} - Deletes a specific information on the logged in user's person.
- **PUT** /api/nsk/v1/persons/{personKey}/merge - Updates a person by merging the PNRs associated, account information and optionally the customer programs.
- **GET** /api/nsk/v1/persons/{personKey}/phoneNumbers - Gets the collection of phone numbers for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/phoneNumbers - Creates a new phone number for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey} - Gets a specific phone number for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey} - Updates a specific phone number for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey} - Patches a specific phone number for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey} - Deletes a specific phone number for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/preferences - Gets all preferences for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/preferences - Creates a new preference for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey} - Gets a specific preference for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey} - Updates a specific preference for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey} - Patches a specific preference for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey} - Deletes a specific preference for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/programs - Gets the collection of programs for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/programs - Creates a new program for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/programs/{personProgramKey} - Gets a specific program for a specific person.
- **PUT** /api/nsk/v1/persons/{personKey}/programs/{personProgramKey} - Updates a specific program for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/programs/{personProgramKey} - Patches a specific program for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/programs/{personProgramKey} - Deletes a specific program for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/storedPayments - Gets the collection of stored payments for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/storedPayments - Creates a new stored payment for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey} - Gets a specific stored payment for a specific person.
- **PATCH** /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey} - Patches a specific stored payment for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey} - Deletes a specific stored payment for a specific person.
- **PUT** /api/nsk/v2/persons/{personKey}/storedPayments/{personStoredPaymentKey} - Updates a specific stored payment for a specific person.

- **GET** /api/nsk/v1/persons/{personKey}/travelDocuments - Gets the collection of travel documents for a specific person.
- **POST** /api/nsk/v1/persons/{personKey}/travelDocuments - Creates a new travel document for a specific person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **GET** /api/nsk/v1/persons/{personKey}/travelDocuments/{personTravelDocumentKey} - Gets a specific travel document for a specific person.
- **DELETE** /api/nsk/v1/persons/{personKey}/travelDocuments/{personTravelDocumentKey} - Deletes a specific travel document for a specific person.
- **PUT** /api/nsk/v2/persons/{personKey}/travelDocuments/{personTravelDocumentKey} - Updates a specific travel document for a specific person.
- **PATCH** /api/nsk/v2/persons/{personKey}/travelDocuments/{personTravelDocumentKey} - Patches a specific travel document for a specific person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications - Gets person travel notifications for a person.
- **POST** /api/nsk/v1/persons/{personKey}/travelNotifications - Creates a person travel notification for a person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey} - Gets a person travel notification for a person.
- **PUT** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey} - Updates a person travel notification for a person.
- **PATCH** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey} - Patches a person travel notification for a person.
- **DELETE** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey} - Deletes a person travel notification for a person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events - Gets notification events for a person travel notification for a person.
- **POST** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events - Creates a notification event for a person travel notification for a person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events/{eventType} - Gets a notification event for a person travel notification for a person.
- **DELETE** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events/{eventType} - Deletes a notification event from a person travel notification for a person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents - Gets notification timed events for a person travel notification for a person.
- **POST** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents - Creates a notification timed event for a person travel notification for a person.
- **GET** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Gets a notification timed event for a person travel notification for a person.
- **PUT** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Updates a notification timed event for a person travel notification for a person.
- **DELETE** /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Deletes a notification timed event from a person travel notification for a person.

Pricing (2 endpoints)

- **POST** /api/apo/v1/pricing - Creates a pricing offer in the ancillary pricing options service.
- **GET** /api/apo/v1/pricing/configuration - Gets the pricing configuration data from the ancillary pricing options service.

Promotions (3 endpoints)

- **GET** /api/nsk/v1/promotions - Gets promotions based on data in the request.
- **GET** /api/nsk/v1/promotions/{promotionCode} - Gets a promotion based on the promotion code.
- **GET** /api/nsk/v1/promotions/{promotionCode}/validate - Evaluates a promotion code and optional organization code to determine if the associated promotion is valid or not.

Queues (9 endpoints)

- **GET** /api/nsk/v2/queues/bookings - Gets the list of queues available.
- **DELETE** /api/nsk/v1/queues/bookings/{bookingQueueCode}/items - Empties the booking queue.
- **PUT** /api/nsk/v1/queues/bookings/{bookingQueueCode}/items/{bookingQueueItemKey} - Moves to a different queue.
- **DELETE** /api/nsk/v1/queues/bookings/{bookingQueueCode}/items/{bookingQueueItemKey} - Removes booking queue items from the specified queue.
- **GET** /api/nsk/v2/queues/bookings/{bookingQueueCode}/next - Gets the next item in the booking queue.
- **POST** /api/nsk/v2/queues/bookings/items/{bookingQueueItemKey} - Places the booking queue item back on the queue.
- **GET** /api/nsk/v1/queues/bookings/queueEvents/{queueEventType} - Gets the list of booking queues that match the specified event type.
- **POST** /api/nsk/v1/queues/travel - Creates a new travel queue entry.
- **GET** /api/nsk/v1/queues/travel/{travelQueueCode}/next - Pops the next item in the travel queue.

Redis (3 endpoints)

- **GET** /api/v1/redis - Gets all the cache items that exist in redis.

- **DELETE** /api/v1/redis - Deletes all cache items.
- **DELETE** /api/v1/redis/{name} - Deletes a specific cache item from redis based on the cache item name.

Resources (151 endpoints)

- **GET** /api/nsk/v1/resources/AccountTransactionCodes - Retrieves the collection of account transaction code resources.
- **GET** /api/nsk/v1/resources/AccountTransactionCodes/{transactionCode} - Retrieves the specific account transaction code resource.
- **GET** /api/nsk/v1/resources/AddressTypes - Retrieves the collection of address type resources.
- **GET** /api/nsk/v1/resources/AddressTypes/{addressTypeCode} - Retrieves the specific address type resource.
- **GET** /api/nsk/v1/resources/Areas - Retrieves the collection of area resources.
- **GET** /api/nsk/v1/resources/Areas/{areaCode} - Retrieves the specific area resource.
- **GET** /api/nsk/v1/resources/Areas/{areaCode}/zones - Retrieves the collection of zones resources.
- **GET** /api/nsk/v1/resources/Areas/{areaCode}/zones/{zoneCode} - Retrieves the specific zone resource.
- **GET** /api/nsk/v1/resources/BaggageTypes - Retrieves the collection of baggage type resources.
- **GET** /api/nsk/v1/resources/BaggageTypes/{baggageTypeCode} - Retrieves the specific baggage type resource.
- **GET** /api/nsk/v1/resources/bundles - Retrieves the collection of bundle configuration resources.
- **GET** /api/nsk/v1/resources/bundles/{bundleCode} - Retrieves a specific bundle configuration based on the bundle code.
- **GET** /api/nsk/v1/resources/bundles/applications - Retrieves the collection of bundle application resources.
- **GET** /api/nsk/v1/resources/bundles/rules - Retrieves the collection of bundle rule resources.
- **GET** /api/nsk/v1/resources/bundles/sets - Retrieves the collection of bundle set resources.
- **GET** /api/nsk/v1/resources/bundles/ssrs - Retrieves the collection of bundle ssr configurations associated with the bundle code.
- **GET** /api/nsk/v1/resources/carriers - Retrieves the collection of carrier resources.
- **GET** /api/nsk/v1/resources/carriers/{carrierCode} - Retrieves a single carrier.
- **GET** /api/nsk/v1/resources/Cities - Retrieves the collection of city resources.
- **GET** /api/nsk/v1/resources/Cities/{cityCode} - Retrieves the specific city resource.
- **GET** /api/nsk/v1/resources/ClassOfServices - Retrieves the collection of class of service resources.
- **GET** /api/nsk/v1/resources/ClassOfServices/{classOfServiceCode} - Retrieves the resource.
- **GET** /api/nsk/v1/resources/CommissionRates - Retrieves the collection of commission rate resources.
- **GET** /api/nsk/v1/resources/CommissionRates/{commissionRateCode} - Retrieves the specific commission rate resource.
- **GET** /api/nsk/v1/resources/ContactTypes - Retrieves the collection of contact type resources.
- **GET** /api/nsk/v1/resources/ContactTypes/{contactTypeCode} - Retrieves the specific contact type resource.
- **GET** /api/nsk/v1/resources/contents - Searches the contents based on the search criteria provided.
- **GET** /api/nsk/v1/resources/contents/{contentId} - Retrieves a resource setting and its related information.
- **DELETE** /api/nsk/v1/resources/contents/{contentId} - Deletes a content item.
- **GET** /api/nsk/v1/resources/contents/{contentId}/data - Retrieves resource setting data based on the accept content type.
- **GET** /api/nsk/v2/resources/countries - Retrieves the collection of country resources.
- **GET** /api/nsk/v2/resources/countries/{countryCode} - Retrieves a specific country resource.
- **GET** /api/nsk/v1/resources/cultures - Retrieves the resources.
- **GET** /api/nsk/v1/resources/cultures/{cultureCode} - Retrieves the resource.
- **GET** /api/nsk/v2/resources/currencies - Retrieves the collection of currency resources.
- **GET** /api/nsk/v2/resources/currencies/{currencyCode} - Retrieves the specific currency resource.
- **GET** /api/nsk/v2/resources/customerPrograms - Retrieves the collection of customer program resources.
- **GET** /api/nsk/v2/resources/customerPrograms/{programCode} - Retrieves the specific customer program resource.
- **GET** /api/nsk/v1/resources/CustomerPrograms/{programCode}/levels/{programLevelCode} - Retrieves the specific program level.
- **GET** /api/nsk/v1/resources/Delays - Retrieves the collection of delay resources.
- **GET** /api/nsk/v1/resources/Delays/{delayCode} - Retrieves the specific delay resource.
- **GET** /api/nsk/v1/resources/Departments - Retrieves the collection of department resources.
- **GET** /api/nsk/v1/resources/Departments/{departmentCode} - Retrieves the specific department resource.
- **GET** /api/nsk/v1/resources/distributionOptions - Retrieves the localizations.
- **GET** /api/nsk/v1/resources/DocumentTypes - Retrieves the collection of document type resources.
- **GET** /api/nsk/v1/resources/DocumentTypes/{documentTypeCode} - Retrieves the specific document type resource.
- **GET** /api/nsk/v1/resources/Domains - Retrieves the collection of domain resources.
- **GET** /api/nsk/v1/resources/Domains/{domainCode} - Retrieves the specific domain resource.
- **GET** /api/nsk/v1/resources/EmailTypes - Retrieves the collection of email type resources.
- **GET** /api/nsk/v1/resources/EmailTypes/{emailTypeCode} - Retrieves the specific email type resource.
- **GET** /api/nsk/v1/resources/ExternalRates - Retrieves the collection of external rate resources.
- **GET** /api/nsk/v1/resources/ExternalRates/{rateId} - Retrieves the specific external rate resource.
- **GET** /api/nsk/v1/resources/ExternalSystems - Retrieves the collection of external system resources.
- **GET** /api/nsk/v2/resources/externalSystems/{systemCode} - Retrieves a single external system resource.
- **GET** /api/nsk/v1/resources/fareBasisCodes - Retrieves the list of fare basis codes.
- **GET** /api/nsk/v1/resources/fareDiscounts - Retrieves a collection of fare discount resources.
- **GET** /api/nsk/v1/resources/fareDiscounts/{fareDiscountCode} - Retrieves a specific fare discount resource.
- **GET** /api/nsk/v1/resources/fareSurcharges - Retrieves a collection of fare surcharge resources.
- **GET** /api/nsk/v1/resources/fareSurcharges/{fareSurchargeCode} - Retrieves a specific fare surcharge resource.
- **GET** /api/nsk/v1/resources/FareTypes - Retrieves the collection of fare type resources.

- **GET /api/nsk/v1/resources/FareTypes/{fareTypeCode}** - Retrieves the specific fare type resource.
- **GET /api/nsk/v1/resources/Fees** - Retrieves the collection of fee resources.
- **GET /api/nsk/v1/resources/Fees/{feeCode}** - Retrieves the specific fee resource.
- **GET /api/nsk/v1/resources/fees/{feeCode}/details** - Gets the fee options.
- **GET /api/nsk/v1/resources/genders** - Retrieves the localizations.
- **GET /api/nsk/v1/resources/HotCardReasons** - Retrieves the collection of hot card reason resources.
- **GET /api/nsk/v1/resources/HotCardReasons/{hotCardReasonCode}** - Retrieves the specific hot card reason resource.
- **GET /api/nsk/v1/resources/IssuanceReasons** - Retrieves the collection of issuance reason resources.
- **GET /api/nsk/v1/resources/IssuanceReasons/{issuanceReasonCode}** - Retrieves the specific issuance reason resource.
- **GET /api/nsk/v1/resources/Locations** - Retrieves the collection of location resources.
- **GET /api/nsk/v1/resources/Locations/{locationCode}** - Retrieves the specific location resource.
- **GET /api/nsk/v2/resources/macs** - Retrieves the collection of mac resources.
- **GET /api/nsk/v1/resources/Macs/{macCode}** - Retrieves the specific mac resource.
- **GET /api/nsk/v1/resources/marketRoutes** - Retrieves the collection of market route resources.
- **GET /api/nsk/v2/resources/markets** - Retrieves the collection of market resources.
- **GET /api/nsk/v2/resources/markets/{locationCode}** - Retrieves a specific market resource.
- **GET /api/nsk/v2/resources/markets/{locationCode}/{travelLocationCode}** - Retrieves the resource.
- **GET /api/nsk/v1/resources/MessageTypes** - Retrieves the collection of message type resources.
- **GET /api/nsk/v1/resources/MessageTypes/{messageTypeCode}** - Retrieves the specific message type resource.
- **GET /api/nsk/v1/resources/organizationStatuses** - Retrieves the localizations.
- **GET /api/nsk/v1/resources/PassengerDiscounts** - Retrieves the collection of passenger discount resources.
- **GET /api/nsk/v1/resources/PassengerDiscounts/{passengerDiscountCode}** - Retrieves the specific passenger discount resource.
- **GET /api/nsk/v1/resources/PassengerTypes** - Retrieves the collection of passenger type resources.
- **GET /api/nsk/v1/resources/PassengerTypes/{passengerTypeCode}** - Retrieves the specific passenger type resource.
- **GET /api/nsk/v1/resources/PaymentMethods** - Retrieves the collection of payment method resources.
- **GET /api/nsk/v1/resources/PaymentMethods/{paymentMethodCode}** - Retrieves the specific payment method resource.
- **GET /api/nsk/v1/resources/PersonInformationTypes** - Retrieves the collection of person information type resources.
- **GET /api/nsk/v1/resources/PersonInformationTypes/{personInformationTypeCode}** - Retrieves the specific person information type resource.
- **GET /api/nsk/v1/resources/PhoneTypes** - Retrieves the collection of phone number type resources.
- **GET /api/nsk/v1/resources/PhoneTypes/{phoneTypeCode}** - Retrieves the specific phone number type resource.
- **GET /api/nsk/v1/resources/postalCodes** - Retrieves the collection of postal codes resources.
- **GET /api/nsk/v1/resources/postalCodes/{postalCode}** - Retrieves the postal code resource.
- **GET /api/nsk/v1/resources/printers** - Retrieves a collection of printer resources.
- **GET /api/nsk/v1/resources/printers/{printerCode}** - Retrieves a specific printer resource.
- **GET /api/nsk/v1/resources/ProductClasses** - Retrieves the collection of product class resources.
- **GET /api/nsk/v1/resources/ProductClasses/{productClassCode}** - Retrieves the specific product class resource.
- **GET /api/nsk/v1/resources/PromotionSettingTypes** - Retrieves the collection of promotion setting type resources.
- **GET /api/nsk/v1/resources/PromotionSettingTypes/{promotionSettingsTypeCode}** - Retrieves the specific promotion setting type resource.
- **GET /api/nsk/v1/resources/ProvinceStates** - Retrieves the collection of province state resources.
- **GET /api/nsk/v1/resources/ProvinceStates/{provinceStateCode}** - Retrieves the specific province state resource.
- **GET /api/nsk/v1/resources/queue/categories** - Gets a response with a list of QueueCategories that match the given parameters.
- **GET /api/nsk/v1/resources/queue/categories/{queueCategoryCode}** - Gets a response with a single QueueCategory that match the given parameters.
- **GET /api/nsk/v1/resources/reminderTimes** - Retrieves the collection of reminder time resource that contains valid times for all passenger travel notification timed events.
- **GET /api/nsk/v1/resources/reminderTimes/{timedEventType}** - Retrieves a specific reminder time resource that contains valid times for a passenger travel notification timed event.
- **GET /api/nsk/v2/resources/roles** - Retrieves the role resources.
- **GET /api/nsk/v2/resources/roles/{roleCode}** - Retrieves a single role resource.
- **GET /api/nsk/v1/resources/roles/{roleCode}/settings** - Retrieves the resource.
- **GET /api/nsk/v1/resources/RoleSettingTypes** - Retrieves the collection of role setting resources.
- **GET /api/nsk/v1/resources/RoleSettingTypes/{roleSettingTypeCode}** - Retrieves the specific role setting resource.
- **GET /api/nsk/v1/resources/searchTypes** - Retrieves the localizations.
- **GET /api/nsk/v2/resources/seatPropertyTypes** - Retrieves a list of seat property types.
- **GET /api/nsk/v2/resources/seatPropertyTypes/{seatPropertyTypeCode}** - Retrieves a single seat property type resource.
- **POST /api/nsk/v1/resources/settings** - Retrieves a resource setting and its related information.
- **POST /api/nsk/v1/resources/settings/data** - Retrieves resource setting data based on the accept content type.
- **GET /api/nsk/v1/resources/ShoppingAttributeGroups** - Retrieves the collection of shopping attribute group resources.
- **GET /api/nsk/v1/resources/ShoppingAttributeGroups/{shoppingAttributeGroupCode}** - Retrieves the specific shopping attribute group resource.
- **GET /api/nsk/v1/resources/ShoppingAttributes** - Retrieves the collection of shopping attribute resources.
- **GET /api/nsk/v1/resources/ShoppingAttributes/{shoppingAttributeCode}** - Retrieves the specific shopping attribute resource.
- **GET /api/nsk/v1/resources/singleSignOnProvider** - Retrieves a list of available single sign on providers.
- **GET /api/nsk/v1/resources/singleSignOnProvider/{providerName}** - Retrieves a specific single sign on provider.

- **GET /api/nsk/v1/resources/SsrGroups** - Retrieves the collection of ssr group resources.
- **GET /api/nsk/v1/resources/SsrGroups/{ssrGroupCode}** - Retrieves the specific ssr group resource.
- **GET /api/nsk/v1/resources/SsrNests** - Retrieves the collection of ssr nest resources.
- **GET /api/nsk/v1/resources/SsrNests/{ssrNestCode}** - Retrieves the specific ssr nest resource.
- **GET /api/nsk/v1/resources/SsrRestrictionResults** - Retrieves the collection of ssr restriction result resources.
- **GET /api/nsk/v1/resources/SsrRestrictionResults/{ssrRestrictionResultCode}** - Retrieves the specific ssr restriction result resource.
- **GET /api/nsk/v1/resources/Ssrs** - Retrieves the collection of ssr resources.
- **GET /api/nsk/v1/resources/Ssrs/{ssrCode}** - Retrieves the specific ssr resource.
- **GET /api/nsk/v1/resources/StandByPriorities** - Retrieves the collection of standby priority resources.
- **GET /api/nsk/v1/resources/StandByPriorities/{standByPriorityCode}** - Retrieves the specific standby priority resource.
- **GET /api/nsk/v1/resources/StationCategories** - Retrieves the collection of station category resources.
- **GET /api/nsk/v1/resources/StationCategories/{stationCategoryCode}** - Retrieves the specific station category resource.
- **GET /api/nsk/v1/resources/Stations** - Retrieves the collection of station resources.
- **GET /api/nsk/v1/resources/Stations/{stationCode}** - Retrieves the specific station resource.
- **GET /api/nsk/v1/resources/stations/{stationCode}/details** - Get the station details.
- **GET /api/nsk/v1/resources/stations/category/{stationCategoryCode}** - Gets a response with stations filtered by category.
- **GET /api/nsk/v1/resources/Suffixes** - Retrieves the collection of suffix resources.
- **GET /api/nsk/v1/resources/Suffixes/{suffixCode}** - Retrieves the specific suffix resource.
- **GET /api/dcs/v1/resources/tailNumbers** - Retrieves the tail number inventory.
- **GET /api/nsk/v1/resources/TimeZones** - Retrieves the collection of time zone resources.
- **GET /api/nsk/v1/resources/TimeZones/{timeZoneCode}** - Retrieves the specific time zone resource.
- **GET /api/nsk/v1/resources/Titles** - Retrieves the collection of title resources.
- **GET /api/nsk/v1/resources/Titles/{titleKey}** - Retrieves the specific title resource.
- **GET /api/nsk/v1/resources/TravelClasses** - Retrieves the collection of travel class resources.
- **GET /api/nsk/v1/resources/TravelClasses/{travelClassCode}** - Retrieves the specific travel class resource.
- **GET /api/nsk/v1/resources/travelerNotifications** - Retrieves a collection of traveler notification resource.
- **GET /api/nsk/v1/resources/travelerNotifications/{notificationTypeCode}** - Retrieves the specified traveler notification resource.
- **GET /api/nsk/v1/resources/travelerNotifications/deliveryMethods** - Retrieves the traveler notification delivery methods filtered by program code and level.
- **GET /api/nsk/v1/resources/travelerNotifications/deliveryMethodSettings** - Retrieves the notification delivery method settings resource.
- **GET /api/nsk/v1/resources/zones/{zoneCode}/subZones** - Retrieves the collection of subzones.
- **GET /api/nsk/v1/resources/zones/{zoneCode}/subZones/{subZoneCode}** - Retrieves the specific subzone.

Seatmaps (2 endpoints)

- **GET /api/nsk/v4/seatmaps/{journeyKey}** - Gets the list of seat maps for a specified journey.
- **GET /api/nsk/v1/seatmaps/{legKey}** - Gets the seat maps for a specified leg.

Seats (2 endpoints)

- **POST /api/dcs/v1/seats/block** - Blocks seats on a specific piece of equipment.
- **DELETE /api/dcs/v1/seats/unblock** - Unblocks seats on a specific piece of equipment.

Settings (31 endpoints)

- **GET /api/nsk/v1/settings/booking** - Get the general settings.
- **GET /api/nsk/v1/settings/booking/checkin** - Get the checkin settings.
- **GET /api/nsk/v1/settings/booking/contact** - Get the contact settings.
- **GET /api/nsk/v1/settings/booking/customerAccount** - Get the custom account settings.
- **GET /api/nsk/v1/settings/booking/fee** - Get the fee settings.
- **GET /api/nsk/v1/settings/booking/flightSearch** - Get the flight search settings.
- **GET /api/nsk/v1/settings/booking/general** - Gets the non-role based general booking settings.
- **GET /api/nsk/v1/settings/booking/passenger** - Get the passenger settings.
- **GET /api/nsk/v2/settings/booking/payment** - Get the role based booking payment settings.
- **GET /api/nsk/v1/settings/booking/reserveFlights** - Get the reserve flight settings.
- **GET /api/nsk/v1/settings/customerPrograms** - Gets customer program settings.
- **GET /api/nsk/v1/settings/eTickets** - Gets the non-role based e-ticket configuration.
- **GET /api/nsk/v1/settings/externalMessageControls** - Gets the external message control settings.
- **GET /api/nsk/v1/settings/general/applicationLogon** - Retrieves the logon settings.
- **GET /api/nsk/v1/settings/general/codes** - Retrieves the codes settings.
- **GET /api/nsk/v1/settings/general/operations** - Retrieves the operations settings.
- **GET /api/nsk/v1/settings/general/organization** - Gets the organization settings.
- **GET /api/nsk/v1/settings/general/session** - Retrieves the session settings.
- **GET /api/nsk/v1/settings/itinerary** - Gets the itinerary settings.
- **GET /api/nsk/v1/settings/loyalty** - Gets the non-role based loyalty settings.
- **GET /api/nsk/v1/settings/notifications/general** - Gets the general New Skies notification settings.

- **GET** /api/nsk/v1/settings/payment - Get the non role based payment settings.
- **GET** /api/nsk/v1/settings/phoneNumberValidation - Gets the non-role based phone number validation settings.
- **GET** /api/nsk/v2/settings/premiumServices - Retrieves the premium services settings.
- **GET** /api/nsk/v1/settings/serviceBundles - Gets the service bundles settings.
- **GET** /api/nsk/v2/settings/skySpeed - Retrieves the role-based and non-role based SkySpeed settings.
- **GET** /api/nsk/v1/settings/system/general - Gets the role-based general system settings.
- **GET** /api/nsk/v1/settings/systemConfiguration/finance - Retrieves the session system configuration finance setting.
- **GET** /api/nsk/v1/settings/travelerNotification - Gets the traveler notification settings.
- **GET** /api/nsk/v1/settings/user/agencyCreation - Retrieves the agency creation settings.
- **GET** /api/nsk/v1/settings/user/customerCreation - Retrieves the customer creation settings.

Token (14 endpoints)

- **PUT** /api/auth/v1-beta/token - BETA - Updates the short lived JWT.
- **GET** /api/nsk/v1/token - Get the information about the current token.
- **DELETE** /api/nsk/v1/token - Abandons the active token.
- **PUT** /api/nsk/v2/token - Given a non-null request, upgrades the current session's logged in user. Otherwise, keeps the active token alive.
- **POST** /api/auth/v1-beta/token/anonymous - BETA - Create a short lived anonymous JWT.
- **POST** /api/nsk/v1/token/culture - Sets the culture code for the lifetime of the token.
- **POST** /api/auth/v1/token/multifactor - Register multi-factor authentication options.
- **PUT** /api/auth/v1-beta/token/role - BETA - Updates the short lived JWT with new role permissions.
- **POST** /api/nsk/v1/token/serverTransfer - Transfers the server context from NewSkies and returns a new token.
- **POST** /api/nsk/v1/token/singleSignOn - Creates the general access token that will grant access to the API.
- **PUT** /api/nsk/v1/token/singleSignOn - Given a non-null request, upgrades the current session's logged in user. Otherwise, keeps the active token alive.
- **POST** /api/auth/v1-beta/token/user - BETA - Creates a short lived JWT specific to a user.
- **PUT** /api/auth/v1-beta/token/user - BETA - Updates an anonymous short lived JWT to login as a user.
- **POST** /api/nsk/v1/token/user/person/singleSignOn - Creates a new user, links it with a provider, then logs in as the new user.

Trip (34 endpoints)

- **POST** /api/nsk/v5/trip - Creates a new booking with the provided journeys, contacts, passengers, and SSRs.
- **POST** /api/nsk/v2/trip/info - Gets the search results for the trip information, based on the search criteria.
- **GET** /api/nsk/v1/trip/info/{legKey}/details - Retrieves cabin and various details for the specified leg key.
- **PATCH** /api/nsk/v1/trip/info/{legKey}/status - Stateless operation for updating the trip information status for a specific flight.
- **GET** /api/nsk/v2/trip/info/{legKey}/status - Retrieves the trip status data for the specified leg key.
- **POST** /api/nsk/v1/trip/info/{legKey}/status/comments - Stateless operation for creating operation comments in the trip information status for a specific flight.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/comments/{commentKey} - Stateless operation for updating the operation comments in the trip information status for a specific flight.
- **DELETE** /api/nsk/v1/trip/info/{legKey}/status/comments/{commentKey} - Stateless operation for deleting the operation comments in the trip information status for a specific flight.
- **POST** /api/nsk/v1/trip/info/{legKey}/status/delays - Stateless operation for creating operation delays in the trip information status for a specific flight.
- **PATCH** /api/nsk/v1/trip/info/{legKey}/status/delays/{delayKey} - Stateless operation for patching the operation delays in the trip information status for a specific flight.
- **DELETE** /api/nsk/v1/trip/info/{legKey}/status/delays/{delayKey} - Stateless operation for deleting trip operation delays in the trip information status for a specific flight.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/cancel - Updates the leg status to cancel.
- **PUT** /api/nsk/v2/trip/info/{legKey}/status/operationDetails/status/closeLeg - Updates a leg status closed.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/closePendingLeg - Updates a leg status to closed pending.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/mishap - Secures all passenger information associated with the flight leg clearing and locking the passenger data. This changes the leg status to Mishap.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/openLeg - Updates a leg status to open.
- **PUT** /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/restore - Updates the leg status to normal.
- **POST** /api/nsk/v2/trip/info/legs - Searches for the inventory legs based on the search criteria.
- **GET** /api/nsk/v2/trip/info/legs/simple - Performs a simplified search for trip information by leg, based upon minimal, required data for the search criteria.
- **GET** /api/nsk/v2/trip/info/simple - Performs a simplified search for trip information based upon minimal, required data for the search criteria.
- **POST** /api/nsk/v1/trip/info/status/operationDetails/status/mishapRestore - Restores locked passenger data associated the leg information. This restores the leg status to normal.
- **POST** /api/nsk/v2/trip/move - Moves a journey on the booking in state.
- **POST** /api/nsk/v3/trip/move/availability - Searches for move availability with the full availability search request. This request will allow for total control over the configuration. If a property is not set, the default settings (AvailabilitySettings) is used in its place.

- **GET** /api/nsk/v3/trip/move/availability/{journeyKey} - Searches for availability with the simple move request. The simple search request covers the bare minimum search criteria and lets default settings (AvailabilitySettings) configure the rest.
- **GET** /api/nsk/v2/trip/move/availability/selfService - Self-service search for move availability.
- **POST** /api/nsk/v1/trip/move/irop - Performs an IROP move for the booking in state. This endpoint adds additional IROP move functionality for agents. See /api/nsk/v2/trip/move for IROP functionality not restricted to agents only.
- **POST** /api/nsk/v1/trip/passengers - Creates a new booking with the provided passengers.
- **POST** /api/nsk/v1/trip/passiveSegments - Creates a passive segment and creates a booking if it doesn't already exist.
- **POST** /api/nsk/v4/trip/rebook/availability - Rebook search with the current booking defaults.
- **GET** /api/nsk/v4/trip/rebook/availability/simple - Simple rebook search with the current booking defaults.
- **GET** /api/nsk/v1/trip/schedule - Retrieves the schedule of flights for a market.
- **POST** /api/nsk/v1/trip/schedule/adHoc - Creates an ad hoc flight.
- **POST** /api/nsk/v4/trip/sell - Sells the list of journeys and creates a booking if it doesn't already exist.
- **POST** /api/nsk/v1/trip/standby/availability - Retrieves same-day standby availability relevant to the current state booking.

User (23 endpoints)

- **GET** /api/nsk/v1/user - Retrieves the current logged in user.
- **POST** /api/nsk/v1/user - Creates a user and person (used by anonymous to create non agent logins).
- **PUT** /api/nsk/v1/user - Updates the current logged in user.
- **PATCH** /api/nsk/v1/user - Patches the current logged in user.
- **GET** /api/nsk/v1/user/bookings - Searches the logged in user booking upcoming and past bookings.
- **GET** /api/nsk/v1/user/bookingsByContactCustomerNumber - Searches for any bookings that have the current user as a contact via their customer number.
- **GET** /api/nsk/v1/user/bookingsByPassenger - Searches the logged in user booking upcoming and past bookings.
- **GET** /api/nsk/v1/user/impersonate - Gets the logged in users current session roles state.
- **POST** /api/nsk/v1/user/impersonate - Impersonates a new role for the logged in user.
- **DELETE** /api/nsk/v1/user/impersonate - Resets the logged in users role.
- **POST** /api/nsk/v3/user/password/change - Changes the logged in user's password.
- **GET** /api/nsk/v1/user/roles - Gets all roles from the logged in user.
- **POST** /api/nsk/v1/user/roles - Creates a new role on the logged in user.
- **GET** /api/nsk/v1/user/roles/{userRoleKey} - Gets a specific role from the logged in user.
- **PUT** /api/nsk/v1/user/roles/{userRoleKey} - Updates a specific role on the logged in user.
- **PATCH** /api/nsk/v1/user/roles/{userRoleKey} - Patches a specific role on the logged in user.
- **DELETE** /api/nsk/v1/user/roles/{userRoleKey} - Deletes a specific role on the logged in user.
- **GET** /api/nsk/v1/user/singleSignOnToken - Gets all of the single sign on tokens associated with a specific user.
- **GET** /api/nsk/v1/user/singleSignOnToken/{providerKey} - Gets a single sign on token associated with a specific user.
- **POST** /api/nsk/v1/user/singleSignOnToken/{providerKey} - Links the requested token with the logged-in user.
- **PUT** /api/nsk/v1/user/singleSignOnToken/{providerKey} - Updates an existing token associated with the logged-in user.
- **PATCH** /api/nsk/v1/user/singleSignOnToken/{providerKey} - Patches an existing token associated with the logged-in user.
- **DELETE** /api/nsk/v1/user/singleSignOnToken/{providerKey} - Deletes a token associated with the single sign on provider from the logged-in user.

User/Person (86 endpoints)

- **GET** /api/nsk/v1/user/person - Retrieves the current logged in person.
- **PUT** /api/nsk/v1/user/person - Updates the current logged in person record basic information.
- **PATCH** /api/nsk/v1/user/person - Patches the current logged in person records basic information.
- **GET** /api/nsk/v1/user/person/account - Retrieves the currently logged in person's account and collections.
- **GET** /api/nsk/v2/user/person/account/collection/{accountCollectionKey}/transactions - Retrieves the currently logged in person's account collection transactions..
- **GET** /api/nsk/v2/user/person/account/transactions - Retrieves all transactions for the currently logged in person.
- **GET** /api/nsk/v1/user/person/addresses - Gets all addresses from the logged in users person.
- **POST** /api/nsk/v1/user/person/addresses - Creates a new address on the logged in users person.
- **GET** /api/nsk/v1/user/person/addresses/{personAddressKey} - Gets a specific address from the logged in users person.
- **PUT** /api/nsk/v1/user/person/addresses/{personAddressKey} - Updates a specific address on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/addresses/{personAddressKey} - Patches a specific address on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/addresses/{personAddressKey} - Deletes a specific address on the logged in users person.
- **GET** /api/nsk/v1/user/person/affiliations - Gets all affiliations from the logged in users person.
- **POST** /api/nsk/v1/user/person/affiliations - Creates a new affiliation on the logged in users person.
- **GET** /api/nsk/v1/user/person/affiliations/{personAffiliationKey} - Gets a specific affiliation from the logged in users person.
- **PUT** /api/nsk/v1/user/person/affiliations/{personAffiliationKey} - Updates a specific affiliation on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/affiliations/{personAffiliationKey} - Deletes a specific affiliation on the logged in users person.
- **GET** /api/nsk/v1/user/person/aliases - Gets all aliases from the logged in users person.
- **POST** /api/nsk/v1/user/person/aliases - Creates a new alias on the logged in users person.
- **GET** /api/nsk/v1/user/person/aliases/{personAliasKey} - Gets a specific alias from the logged in users person.

- **PUT** /api/nsk/v1/user/person/aliases/{personAliasKey} - Updates a specific alias on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/aliases/{personAliasKey} - Patches a specific alias on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/aliases/{personAliasKey} - Deletes a specific alias on the logged in users person.
- **GET** /api/nsk/v1/user/person/comments - Gets all comments from the logged in users person.
- **POST** /api/nsk/v1/user/person/comments - Creates a new comment on the logged in users person.
- **GET** /api/nsk/v1/user/person/comments/{personCommentKey} - Gets a specific comment from the logged in users person.
- **PUT** /api/nsk/v1/user/person/comments/{personCommentKey} - Updates a specific comment on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/comments/{personCommentKey} - Patches a specific comment on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/comments/{personCommentKey} - Deletes a specific comment on the logged in users person.
- **GET** /api/nsk/v1/user/person/emails - Gets all email addresses from the logged in user's person.
- **POST** /api/nsk/v1/user/person/emails - Creates a new email address on the logged in user's person.
- **GET** /api/nsk/v1/user/person/emails/{personEmailAddressKey} - Gets a specific email address from the logged in user's person.
- **PUT** /api/nsk/v1/user/person/emails/{personEmailAddressKey} - Updates a specific email address on the logged in user's person.
- **PATCH** /api/nsk/v1/user/person/emails/{personEmailAddressKey} - Patches a specific email address on the logged in user's person.
- **DELETE** /api/nsk/v1/user/person/emails/{personEmailAddressKey} - Deletes a specific email address on the logged in user's person.
- **GET** /api/nsk/v1/user/person/information - Gets all information from the logged in user's person.
- **POST** /api/nsk/v1/user/person/information - Creates a new information on the logged in user's person.
- **GET** /api/nsk/v1/user/person/information/{personInformationKey} - Gets a specific information from the logged in user's person.
- **PUT** /api/nsk/v1/user/person/information/{personInformationKey} - Updates a specific information on the logged in user's person.
- **PATCH** /api/nsk/v1/user/person/information/{personInformationKey} - Patches a specific information on the logged in user's person.
- **DELETE** /api/nsk/v1/user/person/information/{personInformationKey} - Deletes a specific information on the logged in user's person.
- **GET** /api/nsk/v1/user/person/phoneNumbers - Gets all phone numbers from the logged in users person.
- **POST** /api/nsk/v1/user/person/phoneNumbers - Creates a new phone number on the logged in users person.
- **GET** /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey} - Gets a specific phone number from the logged in users person.
- **PUT** /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey} - Updates a specific phone number on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey} - Patches a specific phone number on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey} - Deletes a specific phone number on the logged in users person.
- **GET** /api/nsk/v1/user/person/preferences - Gets all preferences from the logged in users person.
- **POST** /api/nsk/v1/user/person/preferences - Creates a new preference on the logged in users person.
- **GET** /api/nsk/v1/user/person/preferences/{personPreferenceKey} - Gets a specific preference from the logged in users person.
- **PUT** /api/nsk/v1/user/person/preferences/{personPreferenceKey} - Updates a specific preference on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/preferences/{personPreferenceKey} - Patches a specific preference on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/preferences/{personPreferenceKey} - Deletes a specific preference on the logged in users person.
- **GET** /api/nsk/v1/user/person/programs - Gets all programs from the logged in users person.
- **POST** /api/nsk/v1/user/person/programs - Creates a new program on the logged in users person.
- **GET** /api/nsk/v1/user/person/programs/{personProgramKey} - Gets a specific program from the logged in users person.
- **PUT** /api/nsk/v1/user/person/programs/{personProgramKey} - Updates a specific program on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/programs/{personProgramKey} - Patches a specific program on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/programs/{personProgramKey} - Deletes a specific program on the logged in users person.
- **GET** /api/nsk/v1/user/person/storedPayments - Gets all stored payments from the logged in users person.
- **POST** /api/nsk/v1/user/person/storedPayments - Creates a new stored payment on the logged in users person.
- **GET** /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey} - Gets a specific stored payment from the logged in users person.
- **PUT** /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey} - Updates a specific stored payment on the logged in users person.
- **PATCH** /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey} - Patches a specific stored payment on the logged in users person.
- **DELETE** /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey} - Deletes a specific stored payment on the logged in users person.
- **GET** /api/nsk/v1/user/person/travelDocuments - Gets all travel documents from the logged in users person.
- **POST** /api/nsk/v1/user/person/travelDocuments - Creates a new travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **GET** /api/nsk/v1/user/person/travelDocuments/{personTravelDocumentKey} - Gets a specific travel document from the logged in users person.
- **DELETE** /api/nsk/v1/user/person/travelDocuments/{personTravelDocumentKey} - Deletes a specific travel document on the logged in users person.
- **PUT** /api/nsk/v2/user/person/travelDocuments/{personTravelDocumentKey} - Updates a specific travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **PATCH** /api/nsk/v2/user/person/travelDocuments/{personTravelDocumentKey} - Patches a specific travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **GET** /api/nsk/v1/user/person/travelNotifications - Gets person travel notifications for the logged in user's person.
- **POST** /api/nsk/v1/user/person/travelNotifications - Creates a person travel notification for the logged in user's person.

- **GET** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey} - Gets a person travel notification for the logged in user's person.
- **PUT** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey} - Updates a person travel notification for the logged in user's person.
- **PATCH** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey} - Patches a person travel notification for the logged in user's person.
- **DELETE** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey} - Deletes a person travel notification for the logged in user's person.
- **GET** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events - Gets notification events for a person travel notification for the logged in user's person.
- **POST** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events - Creates a notification event for a person travel notification for the logged in user's person.
- **GET** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events/{eventType} - Gets a notification event for a person travel notification for the logged in user's person.
- **DELETE** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events/{eventType} - Deletes a notification event from a person travel notification for the logged in user's person.
- **GET** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents - Gets notification timed events for a person travel notification for the logged in user's person.
- **POST** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents - Creates a notification timed event for a person travel notification for the logged in user's person.
- **GET** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Gets a notification timed event for a person travel notification for the logged in user's person.
- **PUT** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Updates a notification timed event for a person travel notification for the logged in user's person.
- **DELETE** /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Deletes a notification timed event from a person travel notification for the logged in user's person.

Users (103 endpoints)

- **POST** /api/nsk/v1/users - Creates a new user and person (used by agents to create other users).
- **GET** /api/nsk/v2/users - Search user records.
- **POST** /api/nsk/v2/users - Creates a new user for a person (used by agents to create other users).
- **GET** /api/nsk/v1/users/{userKey} - Retrieves the a specific user.
- **PUT** /api/nsk/v1/users/{userKey} - Updates the specific user.
- **PATCH** /api/nsk/v1/users/{userKey} - Patches the current logged in user.
- **DELETE** /api/nsk/v1/users/{userKey} - Delete the user.
- **GET** /api/nsk/v1/users/{userKey}/bookings - Searches the specific user for upcoming and past bookings.
- **GET** /api/nsk/v1/users/{userKey}/bookingsByPassenger - Searches the specific user upcoming and past bookings by passenger.
- **POST** /api/nsk/v1/users/{userKey}/password/change - Changes a specifics user's password.
- **POST** /api/nsk/v1/users/{userKey}/password/reset - Invokes the forgot password reset for a specific user.
- **GET** /api/nsk/v1/users/{userKey}/person - Retrieves the a specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person - Updates the specific user's person record basic information.
- **PATCH** /api/nsk/v1/users/{userKey}/person - Patches the specific user's person records basic information.
- **GET** /api/nsk/v1/users/{userKey}/person/addresses - Gets all addresses from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/addresses - Creates a new address on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey} - Gets a specific address from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey} - Updates a specific address on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey} - Patches a specific address on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey} - Deletes a specific address on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/affiliations - Gets all affiliations from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/affiliations - Creates a new affiliation for the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey} - Gets a specific affiliation from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey} - Updates a specific affiliation for the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey} - Deletes a specific affiliation for the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/aliases - Gets all aliases from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/aliases - Creates a new alias on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey} - Gets a specific alias from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey} - Updates a specific alias on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey} - Patches a specific alias on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey} - Deletes a specific alias on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/comments - Gets all comments from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/comments - Creates a new comment on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey} - Gets a specific comment from the specific user's person.

- **PUT** /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey} - Updates a specific comment on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey} - Patches a specific comment on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey} - Deletes a specific comment on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/emails - Gets all email addresses from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/emails - Creates a new email address on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey} - Gets a specific email address from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey} - Updates a specific email address on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey} - Patches a specific email address on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey} - Deletes a specific email address on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/information - Gets all information from the logged in user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/information - Creates a new information on the logged in user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/information/{personInformationKey} - Gets a specific information from the logged in user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/information/{personInformationKey} - Updates a specific information on the logged in user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/information/{personInformationKey} - Patches a specific information on the logged in user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/information/{personInformationKey} - Deletes a specific information on the logged in user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/phoneNumbers - Gets all phone numbers from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/phoneNumbers - Creates a new phone number on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey} - Gets a specific phone number from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey} - Updates a specific phone number on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey} - Patches a specific phone number on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey} - Deletes a specific phone number on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/preferences - Gets all preferences for a specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/preferences - Creates a new preference for a specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey} - Gets a specific preference for a specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey} - Updates a specific preference for a specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey} - Patches a specific preference for a specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey} - Deletes a specific preference for a specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/programs - Gets all programs from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/programs - Creates a new program on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey} - Gets a specific program from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey} - Updates a specific program on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey} - Patches a specific program on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey} - Deletes a specific program on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/storedPayments - Gets all stored payments from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/storedPayments - Creates a new stored payment on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey} - Gets a specific stored payment from the specific user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey} - Updates a specific stored payment on the specific user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey} - Patches a specific stored payment on the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey} - Deletes a specific stored payment on the specific user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelDocuments - Gets all travel documents from the specific user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/travelDocuments - Creates a new travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **GET** /api/nsk/v1/users/{userKey}/person/travelDocuments/{personTravelDocumentKey} - Gets a specific travel document from the specific user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/travelDocuments/{personTravelDocumentKey} - Deletes a specific travel document on the specific user's person.

- **PUT** /api/nsk/v2/users/{userKey}/person/travelDocuments/{personTravelDocumentKey} - Updates a specific travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **PATCH** /api/nsk/v2/users/{userKey}/person/travelDocuments/{personTravelDocumentKey} - Patches a specific travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications - Gets person travel notifications for a user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/travelNotifications - Creates a person travel notification for a user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey} - Gets a person travel notification for a user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey} - Updates a person travel notification for a user's person.
- **PATCH** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey} - Patches a person travel notification for a user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey} - Deletes a person travel notification for a user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events - Gets notification events for a person travel notification for a user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events - Creates a notification event for a person travel notification for a user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events/{eventType} - Gets a notification event for a person travel notification for a user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events/{eventType} - Deletes a notification event from a person travel notification for a user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents - Gets notification timed events for a person travel notification for a user's person.
- **POST** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents - Creates a notification timed event for a person travel notification for a user's person.
- **GET** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Gets a notification timed event for a person travel notification for a user's person.
- **PUT** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Updates a notification timed event for a person travel notification for a user's person.
- **DELETE** /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType} - Deletes a notification timed event from a person travel notification for a user's person.
- **GET** /api/nsk/v2/users/{userKey}/preferences/skySpeedSettings - Retrieves the default SkySpeed settings user preferences of the specified user.
- **PUT** /api/nsk/v2/users/{userKey}/preferences/skySpeedSettings - Updates the default SkySpeed settings user preferences of the specified user.
- **GET** /api/nsk/v1/users/{userKey}/roles - Gets all roles for a specific user.
- **POST** /api/nsk/v1/users/{userKey}/roles - Creates a new role for a specific user.
- **GET** /api/nsk/v1/users/{userKey}/roles/{userRoleKey} - Gets a specific role for a specific user.
- **PUT** /api/nsk/v1/users/{userKey}/roles/{userRoleKey} - Updates a specific role for a specific user.
- **PATCH** /api/nsk/v1/users/{userKey}/roles/{userRoleKey} - Patches a specific role for a specific user.
- **DELETE** /api/nsk/v1/users/{userKey}/roles/{userRoleKey} - Deletes a specific role for a specific user.
- **GET** /api/nsk/v1/users/byPerson/{personKey} - Retrieves the a specific user by the person key.

Utilities (1 endpoints)

- **GET** /api/nsk/v1/utilities/stationLocalTime/{stationCode} - Retrieves the station local time.

Verifiedtraveldocuments (4 endpoints)

- **PUT** /api/dcs/v1/verifiedTravelDocuments - Updates a list of verified travel documents for a passenger and associated infant.
- **POST** /api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey} - Adds a new verified travel document to a passenger or the associated infant.
- **DELETE** /api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey} - Deletes the verified travel documents for a passenger and associated infant.
- **POST**
/api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey}/documents/{documentKey} - Adds a new verified travel document to a passenger or the associated infant by the document key.

Voucherissuance (2 endpoints)

- **POST** /api/nsk/v1/voucherIssuance - Creates vouchers based upon data in the request.

- **GET /api/nsk/v1/voucherIssuance/{voucherIssuanceKey}** - Gets a voucher issuance based upon the voucher issuance key.

Vouchers (10 endpoints)

- **GET /api/nsk/v2/vouchers** - Gets vouchers based upon data in the request.
- **GET /api/nsk/v1/vouchers/{voucherKey}** - Gets a specific voucher based upon the voucher key.
- **PATCH /api/nsk/v1/vouchers/{voucherKey}** - Changes the voucher status, type, or expiration date.
- **GET /api/nsk/v2/vouchers/byAgent** - Gets a collection of vouchers based on the specified agent information.
- **GET /api/nsk/v1/vouchers/byDate** - Searches for vouchers issued within the given date range.
- **GET /api/nsk/v2/vouchers/byIssuance/{voucherIssuanceKey}** - Gets vouchers associated with a voucher issuance.
- **GET /api/nsk/v2/vouchers/byMarket** - Gets a collection of vouchers based on the market information.
- **GET /api/nsk/v1/vouchers/byReference/{voucherReferenceNumber}** - Gets a specific voucher based upon the voucher reference number.
- **GET /api/nsk/v1/vouchers/configuration** - Performs a shallow call to get all available voucher configurations.
- **GET /api/nsk/v1/vouchers/configuration/{configurationCode}** - Performs a deep call to get a specific voucher configuration based on the configuration code.

Watchlist (2 endpoints)

- **GET /api/nsk/v1/watchList/{watchListKey}/persons** - Searches for a collection of watch list persons.
- **GET /api/nsk/v1/watchList/persons/{watchListPersonKey}** - Gets a specific watch list person by key.

User Guide: How to Use the API

Quick Start Guide

This guide helps you understand how to use the API for common tasks.

Authentication

Before using any endpoints, you need to authenticate with the system.

Account Operations

The following endpoints help you manage account-related operations:

Changes the accounts password.

Endpoint: POST /api/nsk/v3/account/password/change

This API function is recommended for a user who has an expired password. This API function is restricted to the anonymous user only and remains as the anonymous user even if the change password operation completed.

Usage Tips:

- Use this to create or submit new data

Invokes the forgot password reset for a specific account.

Endpoint: POST /api/nsk/v1/account/password/reset

Alternative API function: accountPasswordForgot

Usage Tips:

- Use this to create or submit new data

Retrieves the payment fee for the fee code by providing the account number and payment method code for a manual

form of payment. **Endpoint:** POST /api/nsk/v1/booking/payments/fees/{feeCode}

This API function behaves like a GET API function but is masked as POST.

Alternative API function: paymentFeeManual

Usage Tips:

- Use this to create or submit new data
 - Required information: feeCode
-

Retrieves the payment fee for the fee code by providing the stored payment key for a stored payment.

The stored payment key will be able to provide to retrieve the account number, payment method code and account number id. **Endpoint:** GET /api/nsk/v1/booking/payments/fees/{feeCode}/storedPayment/{storedPaymentKey}

Alternative API function: paymentFeeStored

Usage Tips:

- Use this to retrieve information
 - Required information: feeCode, storedPaymentKey
-

Creates an organization account refund for a payment from

the booking in state. **Endpoint:** POST /api/nsk/v2/booking/payments/refunds/organizationCredit

IMPORTANT: This API function should NOT be called simultaneously using the same session token to avoid performance problems.

Alternative API function: organizations2PaymentRefundAddv2

Usage Tips:

- Use this to create or submit new data
-

Booking Operations

The following endpoints help you manage booking-related operations:

Sets the 'Removed' status for a bag for a booking stateless.

Endpoint: DELETE /api/dcs/v1/baggage

Agent only. Requires the booking to be committed.

Usage Tips:

- Use this to remove or cancel something
-

Adds baggage to a booking stateless.

Endpoint: POST /api/dcs/v2/baggage

Agent only. Requires the booking to be committed.

Usage Tips:

- Use this to create or submit new data
-

Sets the 'Removed' status for a bag for a booking stateless, by leg key.

Endpoint: DELETE /api/dcs/v1/baggage/byLegKey

Agent only. Requires the booking to be committed.

Usage Tips:

- Use this to remove or cancel something
-

Get the current booking in state.

Endpoint: GET /api/nsk/v1/booking

Note: To add a booking comment please use api/nsk/v3/booking.

plaintext



IMPORTANT: This API function should NOT be called simultaneously using the same session token to avoid performance problems.

Usage Tips:

- Use this to retrieve information

Commits stateful changes made and processes the booking.

Endpoint: POST /api/nsk/v3/booking

IMPORTANT: This API function should NOT be called simultaneously using the same session token to avoid performance problems.

plaintext



This API function can be configured to allow for the newly created record locator to be returned in the data you receive back.

Usage Tips:

- Use this to create or submit new data

Availability Operations

The following endpoints help you manage availability-related operations:

Searches for low fare availability with the full low fare search request. This request allows for total control

over the configuration. If a property is not set, the default settings (LowFareAvailabilitySettings) are used in its place. Note low fare will not always match availability requests for certain configurations including sorting by taxes and fees. Endpoint: POST /api/nsk/v3/availability/lowfare

Although this API function behaves like a GET, POST reduces the limitations resulting from URL length issues. When LowFareAvailabilityRequest.

Usage Tips:

- Use this to create or submit new data

Retrieves the low fare availability estimate using a range of dates.

Endpoint: GET /api/nsk/v1/availability/lowfare/estimate

This API function, unless otherwise specified, will request the pricing and availability for a single adult passenger. However, the number of passengers can be set on the request.

Usage Tips:

- Use this to retrieve information

Searches for low fare availability with the simple search request. The simple search request covers the common

search criteria and lets default settings (LowFareAvailabilitySettings) configure the rest. Note low fare will not always match availability requests for certain configurations included sorting by taxes and fees. Endpoint: POST /api/nsk/v3/availability/lowfare/simple

Although this API function behaves like a GET, POST reduces the limitations resulting from URL length issues.

Alternative API function: availabilityLowFareSimplev3

Usage Tips:

- Use this to create or submit new data

Searches for availability with the full availability search request. This request allows for total control over

the configuration. If a property is not set, default values will be set instead. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out the filters on the availability request. Endpoint: POST /api/nsk/v4/availability/search

Note: It is advised to optimize the availability request with passing in at least one or more (ProductClass, TravelClass, FareTypes or ClassesOfService) so the data you receive back can be optimized as possible. This is even more recommended for mobile applications due to the nature of limited network capabilities.

Usage Tips:

- Use this to create or submit new data
-

Searches for availability with the simple availability search request. This request allows for easy setup for one

way and round trip searches. If a property is not set, default values will be set instead just like they would in the full availability request. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out the filters on the availability simple request. **Endpoint:** POST /api/nsk/v4/availability/search/simple

Note: It is advised to optimize the availability request with passing in at least one or more (ProductClass, TravelClass, FareTypes or ClassesOfService) so the data you receive back can be optimized as possible. This is even more recommended for mobile applications due to the nature of limited network capabilities.

Usage Tips:

- Use this to create or submit new data
-

Payment Operations

The following endpoints help you manage payment-related operations:

Gets a specific add-on's allowed payment methods.

Endpoint: GET /api/nsk/v1/booking addons/{addOnKey}/payments

If the add on's payment required flag is set to true, this API function will return the allowed payment options for the add on. If set to false, null will be returned.

Usage Tips:

- Use this to retrieve information
 - Required information: addOnKey
-

Adds a payment to a specific add-on on the booking.

Endpoint: POST /api/nsk/v1/booking addons/{addOnKey}/payments

IMPORTANT: This API function should NOT be called simultaneously using the same session token to avoid performance problems.

Alternative API function: addOnPaymentAdd

Usage Tips:

- Use this to create or submit new data
 - Required information: addOnKey
-

Gets all the available add-ons allowed payment methods for all the add-ons on the booking in state.

Endpoint: GET /api/nsk/v1/booking addons/payments

Alternative API function: addOnPayment

Usage Tips:

- Use this to retrieve information
-

Retrieves the booking payments on the booking in state.

Endpoint: GET /api/nsk/v1/booking/payments

Requires a booking in state.

Usage Tips:

- Use this to retrieve information
-

Creates a new external payment or prepaid payment on the booking in state.

Endpoint: POST /api/nsk/v5/booking/payments

IMPORTANT: This API function should NOT be called simultaneously using the same session token to avoid performance problems.

plaintext



The payment fields are dynamic to the criteria of your payment provider.

Usage Tips:

- Use this to create or submit new data
-

Troubleshooting Guide

Common Error Codes and Solutions

This guide helps you understand and resolve common issues when using the API.

Detailed Endpoint Documentation

Account Endpoints

POST /api/nsk/v3/account/password/change

Summary: Changes the accounts password.

Description

This endpoint is recommended for a user who has an expired password. This endpoint is restricted to the anonymous user only and remains as the anonymous user even if the change password operation completed. For changing the password of a logged-in user, please use /api/nsk/v3/user/password/change.

plaintext



This follows domain level restrictions and could result in a failed change if there is a minimum time requirement between password changes, invalid password length, or invalid character.

Request Details

Request Body

The change password request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Request

Request with username:

```
json
{
  "credentials": {
    "username": "agentUsername123",
    "alternateIdentifier": null,
    "password": "OldPassword123!",
    "domain": "WWW",
    "location": "WWW",
    "loginRole": "WWWA"
  },
  "newPassword": "NewPassword123!"
}
```

Description: Validates the provided credentials then updates that user's password with the provided new one.

Request with alternate ID:

```
json
{
  "credentials": {
    "username": null,
    "alternateIdentifier": "12345678",
    "password": "OldPassword123!",
    "domain": "WWW",
    "location": "WWW",
    "loginRole": "WWWA"
  },
  "newPassword": "NewPassword123!"
}
```

Description: Functionally the same, but uses alternate IDs such as customer number.

Response 400

Error response for incorrect domain, username, or password.:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:AgentAuthentication: The agent (WWW/ABC123) was not authenticated",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode.

Error response for a locked account.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:AgentLocked: The agent (WWW/ABC123) is locked.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned in the response body."
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode.

Error response for minimum password time.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server>PasswordMinimumTime: The agent (WWW/ABC123) cannot change their password before the Minimum Time is reached.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned in the response body."
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode. Minimum time is determined by domain configuration.

Error response for password policy violation.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server>PasswordPolicy: X, Y, Z",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned in the response body."
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode. Password policy is configurable and determined by the domain.

Error response when the new password is already used as password before.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:PasswordHistory",
      "message": "nsk-server:PasswordHistory",
      "type": "Information",
      "details": null,
      "rawMessage": "The agent (WW/tst01) cannot reuse this password.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    "data": null
  }
}
```

Description: Only returns raw message in debug mode. Unique password count is determined by domain configuration.

POST /api/nsk/v1/account/password/reset

Summary: Invokes the forgot password reset for a specific account.

Description

Request Details

Request Body

The forgot password request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

202 Response

Ok.

Apismessage Endpoints

POST /api/nsk/v1/apisMessage

Summary: Takes the ApisMessage Request and sends it to core to queue a new message. Returns 201 in the response if message successfully queued and a message with Verified and Infant Counts, and 400 if errors occur (And failure text (if any) will be returned).

Description

Request Details

Request Body

The generate APIS Message Request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

400 Response

201 Response

The APIS Message Created Response.

Apo Endpoints

GET /api/nsk/v1/apo

Summary: Gets the current saved ancillary pricing options data.

Description

Request Details

Responses

200 Response

The ancillary pricing options context.

404 Response

POST /api/nsk/v1/apo

Summary: Adds new entries to the saved ancillary pricing options data.

Description

Request Details

Request Body

The input parameters to add.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT /api/nsk/v1/apo

Summary: Adds new or updates existing ancillary pricing options data.

Description

No description available

Request Details

Request Body

The input parameters to add or update.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/apo

Summary: Deletes all of the saved ancillary pricing options data.

Description

Request Details

Responses

200 Response

PUT /api/nsk/v1/apo/{inputParameterKey}

Summary: Updates an individual saved ancillary pricing option's data.

Description

This will replace the current saved ancillary pricing option's value with the given input parameter value, based on the input parameter key.

Request Details

Parameters

Name	Location	Type	Required	Description
inputParameterKey	path	string	Yes	The input parameter key to update.
inputParameterValue	query	string	No	The input parameter value to update.

Responses

200 Response

500 Response

DELETE /api/nsk/v1/apo/{inputParameterKey}

Summary: Deletes a single saved ancillary pricing option's data.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
inputParameterKey	path	string	Yes	The input parameter key to delete.

Responses

200 Response

Availability Endpoints

POST /api/nsk/v3/availability/lowfare

Summary: Searches for low fare availability with the full low fare search request. This request allows for total control over the configuration. If a property is not set, the default settings (LowFareAvailabilitySettings) are used in its place. Note low fare will not always match availability requests for certain configurations including sorting by taxes and fees.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues. When LowFareAvailabilityRequest.GetAllDetails is set to true, New Skies sets the passenger count in the request to zero. This means that passenger count restrictions on a fare are not considered and that any available fare is returned.

Request Details

Request Body

The low fare availability request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The low fare response v2

404 Response

Examples

Response 200

200 OK Response With Macs, One Way:

json



```
{
  "data": {
    "trips": [
      {
        "lowFareAvailableByMarket": {
          "SLC|BOS": [
            {
              "lowestFareAmount": {
                "fareAmount": 40.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 40.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [
                    "E"
                  ],
                  "arrivalTime": "2024-02-01T00:00:00",
                  "departureTime": "2024-02-01T00:00:00",
                  "legs": [
                    {
                      "arrivalTime": "2024-02-01T00:00:00",
                      "departureTime": "2024-02-01T00:00:00",
                      "destination": "BOS",
                      "origin": "SLC",
                      "flightNumber": "5600",
                      "carrierCode": "NV",
                      "equipmentType": "380",
                      "operatingCarrier": null
                    }
                  ],
                  "availableCount": 24,
                  "productClass": null,
                  "referenceFareKey": "MH5Ffn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flg-",
                  "soldOut": false
                }
              ],
              "destination": "BOS",
              "origin": "SLC",
              "departureDate": "2024-02-01T00:00:00",
              "noFlights": false
            }
          ],
          "SLC|MHT": [
            {
              "lowestFareAmount": {
                "fareAmount": 50.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 50.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [

```

```
        "Y"
    ],
    "arrivalTime": "2024-02-01T00:00:00",
    "departureTime": "2024-02-01T00:00:00",
    "legs": [
        {
            "arrivalTime": "2024-02-01T00:00:00",
            "departureTime": "2024-02-01T00:00:00",
            "destination": "MHT",
            "origin": "SLC",
            "flightNumber": "5200",
            "carrierCode": "NV",
            "equipmentType": "280",
            "operatingCarrier": null
        }
    ],
    "availableCount": 24,
    "productClass": null,
    "referenceFareKey": "MH5Ffn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flt",
    "soldOut": false
},
],
"destination": "MHT",
"origin": "SLC",
"departureDate": "2024-02-01T00:00:00",
"noFlights": false
}
]
}
],
"includeTaxesAndFees": false,
"currencyCode": "USD"
}
}
```

Description: Example response for a successful call for a one way flight, with macs included.

200 OK Response, round trip:

json



```
{
  "data": {
    "trips": [
      {
        "lowFareAvailableByMarket": {
          "SLC|BOS": [
            {
              "lowestFareAmount": {
                "fareAmount": 40.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 40.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [
                    "E"
                  ],
                  "arrivalTime": "2024-02-01T00:00:00",
                  "departureTime": "2024-02-01T00:00:00",
                  "legs": [
                    {
                      "arrivalTime": "2024-02-01T00:00:00",
                      "departureTime": "2024-02-01T00:00:00",
                      "destination": "BOS",
                      "origin": "SLC",
                      "flightNumber": "5600",
                      "carrierCode": "NV",
                      "equipmentType": "380",
                      "operatingCarrier": null
                    }
                  ],
                  "availableCount": 24,
                  "productClass": null,
                  "referenceFareKey": "MH5Fn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flg-",
                  "soldOut": false
                }
              ],
              "destination": "BOS",
              "origin": "SLC",
              "departureDate": "2024-02-01T00:00:00",
              "noFlights": false
            }
          ]
        }
      },
      {
        "lowFareAvailableByMarket": {
          "BOS|SLC": [
            {
              "lowestFareAmount": {
                "fareAmount": 80.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 80.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

```

        "taxesAndFeesAmount": 0.0
    }
},
"bookingClasses": [
    "E"
],
"arrivalTime": "2024-02-06T00:00:00",
"departureTime": "2024-02-06T00:00:00",
"legs": [
    {
        "arrivalTime": "2024-02-06T00:00:00",
        "departureTime": "2024-02-06T00:00:00",
        "destination": "SLC",
        "origin": "BOS",
        "flightNumber": "6500",
        "carrierCode": "NV",
        "equipmentType": "830",
        "operatingCarrier": null
    }
],
"availableCount": 50,
"productClass": null,
"referenceFareKey": "MH5FFn50Vn5FULBUU0ZSRX55UFRTfn4wfj",
"soldOut": false
}
],
"destination": "SLC",
"origin": "BOS",
"departureDate": "2024-02-06T00:00:00",
"noFlights": false
}
]
}
],
"includeTaxesAndFees": false,
"currencyCode": "USD"
}
}

```

Description: Example response for a successful call round trip

GET /api/nsk/v1/availability/lowfare/estimate

Summary: Retrieves the low fare availability estimate using a range of dates.

Description

This endpoint, unless otherwise specified, will request the pricing and availability for a single adult passenger. However, the number of passengers can be set on the request. It is important to note that the pricing on this endpoint is estimated. It will not match availability due to round trip vs one way discounts, taxes and fees variations, multi passenger discounts, recently sold out fares, etc. It's intended to be an estimate only. Due to these restrictions this endpoint is extremely performant and can be configured to be cached, with a high cache hit rate.

Request Details

Parameters

Name	Location	Type	Required	Description
StartDate	query	string (date-time)	No	The start date to retrieve from the low fare cache.

plaintext



|

| EndDate | query | string (date-time) | No | The end date for the low fare cache request. || Origin | query | string | No | The origin station code. || Destination | query | string | No | The destination station code. || CurrencyCode | query | string | No | The currency code. || IncludeTaxesAndFees | query | boolean | No | A flag to indicate if taxes and fees should be included. || NumberOfPassengers | query | integer | No | The number of passengers |

Responses

200 Response

404 Response

POST /api/nsk/v3/availability/lowfare/simple

Summary: Searches for low fare availability with the simple search request. The simple search request covers the common search criteria and lets default settings (LowFareAvailabilitySettings) configure the rest. Note low fare will not always match availability requests for certain configurations included sorting by taxes and fees.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Request Body

The low fare availability search simple request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The low fare response.

404 Response

Examples

Response 200

200 OK Response With Macs, One Way:

json



```
{
  "data": {
    "trips": [
      {
        "lowFareAvailableByMarket": {
          "SLC|BOS": [
            {
              "lowestFareAmount": {
                "fareAmount": 40.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 40.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [
                    "E"
                  ],
                  "arrivalTime": "2024-02-01T00:00:00",
                  "departureTime": "2024-02-01T00:00:00",
                  "legs": [
                    {
                      "arrivalTime": "2024-02-01T00:00:00",
                      "departureTime": "2024-02-01T00:00:00",
                      "destination": "BOS",
                      "origin": "SLC",
                      "flightNumber": "5600",
                      "carrierCode": "NV",
                      "equipmentType": "380",
                      "operatingCarrier": null
                    }
                  ],
                  "availableCount": 24,
                  "productClass": null,
                  "referenceFareKey": "MH5Ffn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flg-",
                  "soldOut": false
                }
              ],
              "destination": "BOS",
              "origin": "SLC",
              "departureDate": "2024-02-01T00:00:00",
              "noFlights": false
            }
          ],
          "SLC|MHT": [
            {
              "lowestFareAmount": {
                "fareAmount": 50.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 50.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [

```

```
        "Y"
    ],
    "arrivalTime": "2024-02-01T00:00:00",
    "departureTime": "2024-02-01T00:00:00",
    "legs": [
        {
            "arrivalTime": "2024-02-01T00:00:00",
            "departureTime": "2024-02-01T00:00:00",
            "destination": "MHT",
            "origin": "SLC",
            "flightNumber": "5200",
            "carrierCode": "NV",
            "equipmentType": "280",
            "operatingCarrier": null
        }
    ],
    "availableCount": 24,
    "productClass": null,
    "referenceFareKey": "MH5Ffn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flt",
    "soldOut": false
},
],
"destination": "MHT",
"origin": "SLC",
"departureDate": "2024-02-01T00:00:00",
"noFlights": false
}
]
}
],
"includeTaxesAndFees": false,
"currencyCode": "USD"
}
}
```

Description: Example response for a successful call for a one way flight, with macs included.

200 OK Response, round trip:

json



```
{
  "data": {
    "trips": [
      {
        "lowFareAvailableByMarket": {
          "SLC|BOS": [
            {
              "lowestFareAmount": {
                "fareAmount": 40.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 40.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  },
                  "bookingClasses": [
                    "E"
                  ],
                  "arrivalTime": "2024-02-01T00:00:00",
                  "departureTime": "2024-02-01T00:00:00",
                  "legs": [
                    {
                      "arrivalTime": "2024-02-01T00:00:00",
                      "departureTime": "2024-02-01T00:00:00",
                      "destination": "BOS",
                      "origin": "SLC",
                      "flightNumber": "5600",
                      "carrierCode": "NV",
                      "equipmentType": "380",
                      "operatingCarrier": null
                    }
                  ],
                  "availableCount": 24,
                  "productClass": null,
                  "referenceFareKey": "MH5Fn50Vn5FU1BUU0ZSRX55UFRTfn4wfjF_flg-",
                  "soldOut": false
                }
              ],
              "destination": "BOS",
              "origin": "SLC",
              "departureDate": "2024-02-01T00:00:00",
              "noFlights": false
            }
          ]
        }
      },
      {
        "lowFareAvailableByMarket": {
          "BOS|SLC": [
            {
              "lowestFareAmount": {
                "fareAmount": 80.0,
                "farePointAmount": 0.0,
                "taxesAndFeesAmount": 0.0
              },
              "lowFares": [
                {
                  "passengers": {
                    "adt": {
                      "discountCode": null,
                      "fareAmount": 80.0,
                      "type": "ADT",
                      "farePointAmount": 0.0,
                      "taxesAndFeesAmount": 0.0
                    }
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

```

        "taxesAndFeesAmount": 0.0
    }
},
"bookingClasses": [
    "E"
],
"arrivalTime": "2024-02-06T00:00:00",
"departureTime": "2024-02-06T00:00:00",
"legs": [
    {
        "arrivalTime": "2024-02-06T00:00:00",
        "departureTime": "2024-02-06T00:00:00",
        "destination": "SLC",
        "origin": "BOS",
        "flightNumber": "6500",
        "carrierCode": "NV",
        "equipmentType": "830",
        "operatingCarrier": null
    }
],
"availableCount": 50,
"productClass": null,
"referenceFareKey": "MH5Ffn50Vn5FULBUU0ZSRX5SUFRFn4wfj",
"soldOut": false
}
],
"destination": "SLC",
"origin": "BOS",
"departureDate": "2024-02-06T00:00:00",
"noFlights": false
}
]
}
],
"includeTaxesAndFees": false,
"currencyCode": "USD"
}
}

```

Description: Example response for a successful call round trip

POST /api/nsk/v4/availability/search

Summary: Searches for availability with the full availability search request. This request allows for total control over the configuration. If a property is not set, default values will be set instead. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out the filters on the availability request.

Description

Note: It is advised to optimize the availability request with passing in at least one or more (ProductClass, TravelClass, FareTypes or ClassesOfService) so the response can be optimized as possible. This is even more recommended for mobile applications due to the nature of limited network capabilities. The default values that will be set unless specified:

plaintext

```
- TaxesAndFees = TaxesAndFeesRollupMode.None
- Criteria.SsrCollectionsMode = AvailabilitySsrCollectionsMode.None
- Criteria.Filters.FareInclusionType = AvailabilityType.Default
- Criteria.Filters.CompressionType = FareClassControl.LowestFareClass
- Criteria.Filters.Loyalty = LoyaltyFilter.MonetaryOnly
- Criteria.Filters.FlightType = FlightType.All
- Criteria.Filters.ExclusionType = AvailabilityFilter.ExcludeUnavailable
- Criteria.Filters.SortOptions =
JourneySortKey.LowestFare,JourneySortKey.EarliestDeparture,JourneySortKey.ShortestTravelTime
- Filters.ConnectionType = SoldAsConnectionType.None
- Criteria.Filters.MaxConnectingFlights = 0
- Criteria.Stations.SearchDestinationMacs = false
- Criteria.Stations.SearchOriginMacs = false
- Criteria.Filters.BundleControlFilter = BundleControlFilter.Disabled
- NumberOfFaresPerJourney = 1
```

Criteria.Filters.IncludeServiceBundleSetCodes is marked as obsolete, and the use of BundleControlFilter is preferred. To specifically return bundle set codes which IncludeServiceBundleSetCodes has originally provided when set to true, use an enumeration value of BundleControlFilter.ReturnBundleSets for BundleControlFilter instead. When searching for open jaw availability the results will contain only available markets if any are configured. The AvailabilityCriteriaFilter.FareInclusionType.NoPricing enumeration value, if set, will result in an availability response that does not include fares. The primary use case for this is in Move flows, but is being maintained on the main availability and rebook availability flows in case there are other use cases as well.

Request Details

Request Body

The availability search request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

Examples

Response 200

Requested Passenger Discount Overridden Message:

json

```
{
  "messages": [
    {
      "rawValue": "A passenger discount code was found that is associated with the requested promotion code. The passenger discount code is \"PassengerDiscountCodeOverridden\". This message is being returned in debug mode. In production the 'rawValue' node will not be returned. These are informational messages and do not affect the availability data returned by the API.",
      "code": "PassengerDiscountCodeOverridden",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are informational messages and do not affect the availability data returned by the API.",
      "type": "Information",
      "value": "PassengerDiscountCodeOverridden",
      "status": 0,
      "details": {
        "associatedDiscountCode": "SNR"
      }
    }
  ],
  "data": null
}
```

Description: Example OSI Message when getting availability with a promotion code, no requested passenger discount, and the promotion code has a passenger discount associated with it. **NOTE:**This example *only* displays the message, however, the availability data will be populated in addition to the given message.

POST /api/nsk/v4/availability/search/simple

Summary: Searches for availability with the simple availability search request. This request allows for easy setup for one way and round trip searches. If a property is not set, default values will be set instead just like they would in the full availability request. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out the filters on the availability simple request.

Description

Note: It is advised to optimize the availability request with passing in at least one or more (ProductClass, TravelClass, FareTypes or ClassesOfService) so the response can be optimized as possible. This is even more recommended for mobile applications due to the nature of limited network capabilities. The default values that will be set unless specified:

plaintext

```
- TaxesAndFees = TaxesAndFeesRollupMode.None  
- SsrCollectionsMode = AvailabilitySsrCollectionsMode.None  
- Filters.FareInclusionType = AvailabilityType.Default  
- Filters.CompressionType = FareClassControl.LowestFareClass  
- Filters.Loyalty = LoyaltyFilter.MonetaryOnly  
- Filters.FlightType = FlightType.All  
- Filters.ExclusionType = AvailabilityFilter.ExcludeUnavailable  
- Filters.SortOptions =  
JourneySortKey.LowestFare, JourneySortKey.EarliestDeparture, JourneySortKey.ShortestTravelTime  
- Filters.ConnectionType = SoldAsConnectionType.None  
- Filters.MaxConnectingFlights = 0  
- SearchDestinationMacs = false  
- SearchOriginMacs = false  
- Criteria.Filters.BundleControlFilter = BundleControlFilter.Disabled  
- NumberOfFaresPerJourney = 1
```

Note that the searchOriginMacs and searchDestinationMacs flags apply to both journeys when booking a round-trip journey with the simple availability endpoint. For example, if you set "searchOriginMacs" to "true" for a round trip from SLC-BOS and back, it will search for SLC MAC airports on the outbound (SLC > BOS) journey, and BOS MAC airports on the inbound (BOS > SLC) journey

Criteria.Filters.IncludeServiceBundleSetCodes is marked as obsolete, and the use of BundleControlFilter is preferred. To specifically return bundle set codes which IncludeServiceBundleSetCodes has originally provided when set to true, use an enumeration value of BundleControlFilter.ReturnBundleSets for BundleControlFilter instead. The AvailabilityCriteriaFilter.FareInclusionType.NoPricing enumeration value, if set, will result in an availability response that does not include fares. The primary use case for this is in Move flows, but is being maintained on the main availability and rebook availability flows in case there are other use cases as well.

Request Details

Request Body

The availability simple v2 search request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The availability v2 response.

404 Response

Examples

Response 200

Requested Passenger Discount Overridden Message:

```

json
{
  "messages": [
    {
      "rawValue": "A passenger discount code was found that is associated with the requested promotion code. The passenger discount code is \"PassengerDiscountCodeOverridden\". The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are informational messages.",
      "code": "PassengerDiscountCodeOverridden",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are informational messages.",
      "type": "Information",
      "value": "PassengerDiscountCodeOverridden",
      "status": 0,
      "details": {
        "associatedDiscountCode": "SNR"
      }
    },
    "data": null
  ]
}

```

Description: Example OSI Message when getting availability with a promotion code, no requested passenger discount, and the promotion code has a passenger discount associated with it. *NOTE:*This example *only* displays the message, however, the availability data will be populated in addition to the given message.

POST /api/nsk/v2/availability/search/ssr

Summary: Searches for availability with SSR prices with the full availability search request. This request allows for total control over the configuration. If a property is not set, default values will be set instead. These defaults are what we consider a general optimized request. If these optimizations are not preferred the requester will need to opt out of the filters on the availability request.

Description

Note: This is an expensive call and should be used with appropriate cautions. It is advised to optimize the availability request with passing in at least one or more (ProductClass, TravelClass, FareTypes or ClassesOfService) so the response can be optimized as possible. This is even more recommended for mobile applications due to the nature of limited network capabilities. The default values that will be set unless specified:

```

plaintext
-
- AvailabilityWithSsrRequest.TaxesAndFees = TaxesAndFeesRollupMode.None
- AvailabilityWithSsrRequest.Criteria.SsrCollectionsMode = AvailabilitySsrCollectionsMode.None
- AvailabilityWithSsrRequest.Criteria.Filters.FareInclusionType = AvailabilityType.Default
- AvailabilityWithSsrRequest.Criteria.Filters.CompressionType = FareClassControl.LowestFareClass
- AvailabilityWithSsrRequest.Criteria.Filters.Loyalty = LoyaltyFilter.MonetaryOnly
- AvailabilityWithSsrRequest.Criteria.Filters.FlightType = FlightType.All
- AvailabilityWithSsrRequest.Criteria.Filters.ExclusionType = AvailabilityFilter.ExcludeUnavailable
- AvailabilityWithSsrRequest.Criteria.Filters.SortOptions =
  JourneySortKey.LowestFare, JourneySortKey.EarliestDeparture, JourneySortKey.ShortestTravelTime
- AvailabilityWithSsrRequest.Filters.ConnectionType = SoldAsConnectionType.None
- AvailabilityWithSsrRequest.Criteria.Filters.MaxConnectingFlights = 0
- AvailabilityWithSsrRequest.Criteria.Stations.SearchDestinationMacs = false
- AvailabilityWithSsrRequest.Criteria.Stations.SearchOriginMacs = false
- AvailabilityWithSsrRequest.Criteria.Filters.BundleControlFilter = BundleControlFilter.Disabled
- AvailabilityWithSsrRequest.NumberOfFaresPerJourney = 1

```

AvailabilityWithSsrRequest.Criteria.Filters.IncludeServiceBundleSetCodes is marked as obsolete, and the use of BundleControlFilter is preferred. To specifically return bundle set codes which IncludeServiceBundleSetCodes has originally provided when set to true, use an enumeration value of BundleControlFilter.ReturnBundleSets for BundleControlFilter instead. The AvailabilityCriteriaFilter.FareInclusionType.NoPricing enumeration value, if set, will result in an availability response that does not include fares. The primary use case for this is in Move flows, but is being maintained on the main availability and rebook availability flows in case there are other use cases as well.

Request Details

Request Body

The availability with SSR search request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

Baggage Endpoints

DELETE /api/dcs/v1/baggage

Summary: Sets the 'Removed' status for a bag for a booking stateless.

Description

Agent only. Requires the booking to be committed.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

A request to set bag status as 'Removed' for the provided passenger bag.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

POST /api/dcs/v2/baggage

Summary: Adds baggage to a booking stateless.

Description

Agent only. Requires the booking to be committed. If you have the booking in state you will need to re-retrieve it to see the added baggage.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

A request to add bags to the given passenger.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

400 Response

Adding bag failed.

Examples

Response 201

Add Baggage Successful:

```
json   
{  
    "data": null  
}
```

Description: Example when baggage is added successfully

Response 400

Invalid Record Locator Error:

```
json   
{  
    "errors": [  
        {  
            "id": "2f5dd7f3-47f7-48fb-a180-f9152278605f",  
            "code": "nsk:Exceptions:InvalidKey",  
            "message": "nsk:Exceptions:InvalidKey",  
            "type": "Validation",  
            "details": {  
                "recordLocator": "9SkFNRegdQRYuQ3Yfa2g9"  
            },  
            "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3Yfa2g9' is invalid.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        },  
        ],  
        "data": null  
    }  
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json   
{  
    "errors": [  
        {  
            "id": "2f5dd7f3-47f7-48fb-a180-f9152278605f",  
            "code": "nsk:Exceptions:InvalidKey",  
            "message": "nsk:Exceptions:InvalidKey",  
            "type": "Validation",  
            "details": {  
                "passengerKey": "JoKwkYODu4gNYRwNQo"  
            },  
            "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        },  
        ],  
        "data": null  
    }  
}
```

Description: Example error when passenger key is invalid.

Invalid Journey Key Error:

```
json
{
  "errors": [
    {
      "id": "2f5dd7f3-47f7-48fb-a180-f9152278605f",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "journeyKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'JourneyKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when journey key is invalid.

Invalid Baggage Type Code Error:

```
json
{
  "errors": [
    {
      "id": "1a68ef3e-44b5-08d4-9395-e69695467a5d",
      "code": "dcs:Baggage:InvalidBaggageTypeCode",
      "message": "dcs:Baggage:InvalidBaggageTypeCode",
      "type": "Validation",
      "details": null,
      "rawMessage": "The baggage type code 'BA20' is not valid or is inactive for the current environment.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when baggage type code is invalid.

Add Baggage Failure Without Specific Message:

```
json
{
  "errors": [
    {
      "id": "693b6295-6ae8-a005-9f1e-1e1d1600d573",
      "code": "dcs:Exceptions>CreateFailed",
      "message": "dcs:Exceptions>CreateFailed",
      "type": "Validation",
      "details": null,
      "rawMessage": "Attempted create of object 'AddBaggageRequest' has failed. Please verify this object does not already exist.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when add baggage fails in New Skies without a specific error message.

Add Baggage Failure With Specific Message:

```
json

{
  "errors": [
    {
      "id": "693b6295-6ae8-a005-9f1e-1e1d1600d573",
      "code": "dcs:Exceptions:CreateFailed",
      "message": "Flight 2241 SLCBOS has been cancelled, request not processed.",
      "type": "Error",
      "details": {
        "journeyKey": "T1Z_NTYxNH4gfn5TTEN_MDYvMjgvMjAyNCAwMzowMH5ERU5_MDYvMjgvMjAyNCAwNDowMH5_Xk5Wfju2MTV_IH5_REVOfjA2LzI4LzIwMjO",
        "passengerKey": "MCFBRFQ_",
        "errorMessage": "Flight 2241 SLCBOS has been cancelled, request not processed."
      },
      "rawMessage": "Flight 2241 SLCBOS has been cancelled, request not processed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Example error when add baggage fails in New Skies with a specific error message.

DELETE /api/dcs/v1/baggage/byLegKey

Summary: Sets the 'Removed' status for a bag for a booking stateless, by leg key.

Description

Agent only. Requires the booking to be committed.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance
problems.
All downline legs and segments will have the bag status marked as `Removed` as well.
If you want to remove the bag for a whole journey, just pass in the first leg key.
```

Request Details

Request Body

A request to set bag status as 'Removed' for the provided passenger bag.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

409 Response

Concurrent changes were being made in the same session.

Examples

Response 400

Invalid passenger key error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'PassengerKey' with value 'tW' is invalid.",
      "type": "Validation",
      "details": {
        "passengerKey": "tW"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when passenger key is invalid.

Invalid leg key error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'LegKey' with value '76oyA0KoZ0gsjHX' is invalid.",
      "type": "Validation",
      "details": {
        "legKey": "76oyA0KoZ0gsjHX"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when leg key is invalid.

Invalid baggage key error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'BaggageKey' with value 'upvbql' is invalid.",
      "type": "Validation",
      "details": {
        "baggageKey": "upvbql"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when baggage key is invalid.

Invalid record locator error:

```
json   
  
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "The identifier 'RecordLocator' with value 'n02' is invalid.",  
      "type": "Validation",  
      "details": {  
        "recordLocator": "n02"  
      },  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  "data": null  
}
```

Description: Example error when the record locator invalid.

PUT /api/dcs/v1/baggage/checkIn

Summary: Performs an update and checks-in an individual baggage item for a specific passenger.

Description

If the journey key is set on the request, the request will process all segments associated with the journey, and will ignore the segment key value. The baggage type code can only be updated if the baggage weight is changed as well, and if the baggage type code is a valid, active baggage type in the current environment. Please note, this call does NOT sync with the booking baggage allowances and usages.

```
plaintext   
  
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Request Body

The passenger bag edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

500 Response

Examples

Response 400

Invalid leg status error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Baggage:BaggageCheckinFailed",
      "message": "nsk:Baggage:BaggageCheckinFailed",
      "type": "Validation",
      "details": {
        "segmentKey": "9SkFNRegdQRYuQ3YfA2g9",
        "legKey": "9SkFNReg",
        "flightNumber": "1234",
        "origin": "JFK",
        "destination": "LAX",
        "legStatus": "Closed"
      },
      "rawMessage": "Flight must be open to checkin baggage. One or more legs on the given journey/segment is not open.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when leg status is invalid.

Bagtag Endpoints

GET /api/dcs/v1/bagTag/printers

Summary: Gets the bag tag printers.

Description

This information is retrieved based on the logged in user's location. If a location is not set up, the returned information will be the default printers configured in printers.xml from UTL\Resource Management.

Request Details

Responses

404 Response

200 Response

Boarding Endpoints

POST /api/dcs/v2/boarding/{recordLocator}/legs/{legKey}/passengers/{passengerKey}

Summary: Boards a passenger on a specific leg. If the leg is NOT a part of a change of gauge flight, it is recommended to board by segment instead.

Description

If there are downstream legs that are not boarded on the segment and the flight is not a change of gauge, the rest of the legs will be boarded as well. For example, if you have a three leg segment and you board the middle leg, the last leg will also be boarded and the first leg will not be boarded. This endpoint will return information like a GET because there may be necessary security information on the response. To check the lift status of the boarded leg, please use /api/nsk/manifest/{legKey}.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator for the booking.
legKey	path	string	Yes	The leg key to board.
passengerKey	path	string	Yes	The passenger key.

Request Body

The board passenger request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

DELETE /api/dcs/v1/boarding/{recordLocator}/legs/{legKey}/passengers/{passengerKey}

Summary: Unboards a passenger on a specific leg.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator for the booking.
legKey	path	string	Yes	The leg key to board.
passengerKey	path	string	Yes	The passenger key.

Responses

200 Response

POST

/api/dcs/v2/boarding/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey}

Summary: Boards a passenger on a specific segment. A segment should not be boarded if it is a change of gauge flight.

Description

If an attempt at boarding a segment is made with a change of gauge flight, only the first leg of that segment will be boarded. The endpoint will return that the process was successful, but the full segment with legs was not boarded. This can be checked using the endpoint listed below. This endpoint will return information like a GET because there may be necessary security information on the response. To check the lift status of the boarded segment please use /api/nsk/v1/bookings/checkin/{recordLocator}/segments/{segmentKey}/status. That endpoint can be used after boarding as well because it retrieves the lift status of the segment.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator for the booking.
segmentKey	path	string	Yes	The segment key to board.
passengerKey	path	string	Yes	The passenger key.

Request Body

The board passenger request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

DELETE /api/dcs/v1/boarding/legs/{legKey}/passengers

Summary: Unboards all passengers on a leg.

Description

If a warning message is present, check the lift status of the passengers. A passenger will come back as not being unboarded if they are not checked in and boarded.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

POST /api/dcs/v2/boarding/legs/{legKey}/passengers

Summary: Boards all passengers on a specific leg.

Description

If there are downstream legs that are not boarded on the segment and the flight is not a change of gauge, the rest of the legs will be boarded as well. For example, if you have a three leg segment and you board the middle leg, the last leg will also be boarded and the first leg will not be boarded. If the flight is a change of gauge, then only the leg requested will be boarded. To check the lift status of the boarded leg, please use /api/nsk/manifest/bookings/{recordLocator}/legs/{legKey}.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key to board.

Responses

200 Response

404 Response

Booking Endpoints

GET /api/nsk/v1/booking

Summary: Get the current booking in state.

Description

Note: To add a booking comment please use api/nsk/v3/booking.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

OK

404 Response

Not found.

POST /api/nsk/v3/booking

Summary: Commits stateful changes made and processes the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext



This endpoint can be configured to allow for the newly created record locator to be returned in the response.
This is
an API value that needs to be set in configuration, NskSystemOptions.ReturnPnrOnv3BookingPost = true.
Default behavior is to not return the new record locator. This will require a SNOW ticket to change.

Request Details

Request Body

Commit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Committed successfully.

400 Response

Examples

Response 400

Duplicate passenger names:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:DuplicatePassengerName",
      "message": "Booking is not allowed to have two passengers with the same name: Derek Holmes",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: Example error with 2 passengers with identical names.

Modified Booking:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingWasModified",
      "message": "Booking ZZZ123 has been modified by another user.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: Booking was modified since last viewed, please reload.

Class Not Available:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:ClassNotAvailable",
      "message": "The requested class of service is sold out.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: The class of service has sold out since the shopping request was made.

SSR Not Available:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:SSRNotAvailable",
      "message": "SSR PS is not available on flight FR1123 SLC-DEN for passenger John Smith.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: The SSR has sold out since the ssr availability request was made.

Hold Not Allowed Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:HoldNotAllowed",
      "message": "A booking hold is not allowed for this booking. Verify that the booking has journeys, balance due, or override information defined before attempting to hold it.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when Committing Booking while Hold Date Override is disabled.

Hold Greater Than Allowed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:HoldBookingNotAllowed",
      "message": "The selected hold date is not allowed. The value, 11/14/2023 00:00:00, and is greater than the allowed hold date of 11/13/2023 00:00:00.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when Committing Booking with Hold Date Greater Than allowed.

PUT /api/nsk/v3/booking

Summary: Commits stateful changes made and processes the booking.

Description

Only archived bookings can use the `Archive` comment type. If the booking in state is archived, only the comments will be acknowledged by New Skies. All other filled in fields in the request will be ignored. If `AllowConcurrentChanges` is set to true, a merge between concurrent changes between multiple sessions will be performed before committing the booking. New Skies refers to this as `MergeAndCommit2`. We recommend only setting this flag to true for cases where concurrent booking changes are highly possible (e.g. multiple sessions making changes to the same booking right before checking-in).

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
allowConcurrentChanges	query	boolean	No	Flag to allow concurrent booking changes between multiple sessions.

Request Body

Commit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

500 Response

Examples

Response 200

Hold Date Changed:

json



```
{  
  "messages": [  
    {  
      "rawValue": "The hold date has changed from 10/10/2023 08:50 to 10/25/2023 00:00.",  
      "code": "",  
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",  
      "type": "HoldDateChanged",  
      "value": null,  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": null  
}
```

Description: Example response when hold date is changed.

Response 400

Duplicate passenger names:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:DuplicatePassengerName",
      "message": "Booking is not allowed to have two passengers with the same name: Derek Holmes",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: Example error with 2 passengers with identical names.

Modified Booking:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingWasModified",
      "message": "Booking ZZZ123 has been modified by another user.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: Booking was modified since last viewed, please reload.

Class Not Available:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:ClassNotAvailable",
      "message": "The requested class of service is sold out.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}
```

Description: The class of service has sold out since the shopping request was made.

SSR Not Available:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:SSRNotAvailable",
      "message": "SSR PS is not available on flight FR1123 SLC-DEN for passenger John Smith.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: The SSR has sold out since the ssr availability request was made.

Hold Not Allowed Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:HoldNotAllowed",
      "message": "A booking hold is not allowed for this booking. Verify that the booking has journeys, balance due, or override is enabled before committing the booking.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when Committing Booking while Hold Date Override is disabled.

Hold Greater Than Allowed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:HoldBookingNotAllowed",
      "message": "The selected hold date is not allowed. The value, 11/14/2023 00:00:00, and is greater than the allowed hold date of 11/13/2023 00:00:00.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when Committing Booking with Hold Date Greater Than allowed.

GET /api/nsk/v1/booking/baggageAllowances

Summary: Gets the baggage allowances for the current booking in state. Baggage usage details will only be returned if includeUsageDetails is flagged true AND the allowance behavior is NOT Default.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
includeUsageDetails	query	boolean	No	If true, the baggage allowance usage details will be populated in the response.
Including these details adds processing overhead, so set this only when absolutely necessary.				
Defaults to <code>false</code> , not include usage details.				

```
plaintext 
```

Responses

200 Response

The collection of passenger baggage allowances.

404 Response

POST /api/nsk/v1/booking/baggageAllowances/external

Summary: Gets the third-party baggage allowance calculations for the current booking in state.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Request Body

External baggage allowance request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/bundle/availability

Summary: Retrieves the bundle and srr availability for the booking in-state.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Request Body

The bundle availability request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

404 Response

200 Response

GET /api/nsk/v1/booking/comments

Summary: Gets the list of comments on the in-state booking.

Description

No description available

Request Details

Responses

200 Response

The list booking comments, if any.

404 Response

DELETE /api/nsk/v2/booking/comments/{commentKey}

Summary: Deletes a comment from the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
commentKey	path	string	Yes	The booking comment key.

Responses

200 Response

POST /api/nsk/v2/booking/divide

Summary: Divides the current booking.

Description

A valid user must be logged in to divide a booking.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Contains the divide request information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Successfully divided.

200 Response

Successfully divided and canceled.

GET /api/nsk/v2/booking/equipment/legs/{legKey}

Summary: Retrieves equipment information for a specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

404 Response

GET /api/nsk/v2/booking/equipment/legs/{legKey}/properties

Summary: Retrieves equipment, supported seat property and seat-dependent SSR information for a specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

404 Response

GET /api/nsk/v2/booking/equipment/segments/{segmentKey}

Summary: Retrieves equipment information for all legs on a specified segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.

Responses

200 Response

404 Response

GET /api/nsk/v2/booking/equipment/segments/{segmentKey}/properties

Summary: Retrieves equipment, supported seat property and seat-dependent SSR information for all legs on a specified segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/eTickets

Summary: Issues new e-tickets and re-issues existing e-tickets for segments with voluntary changes.

Description

The endpoint performs validation and determines internally if the booking is for issuance of new e-tickets or re-issuance of existing e-tickets. For some cases that the re-issuance is not allowed by the ticketing service, new e-tickets are issued instead.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

201 Response

Successfully issued or re-issued e-tickets.

502 Response

Examples

Response 201

Successfully issue or reissue new e-tickets:

json



```
{  
  "ticketNumbers": [  
    "0286010103190",  
    "0286010103191"  
  ]  
}
```

Description: Example response when issuance or re-issuance of e-tickets is successful.

PUT /api/nsk/v1/booking/eTickets

Summary: Re-issues existing e-tickets for segments with involuntary changes such as schedule change and IROP.

Description

For some cases that the re-issuance is not allowed by the ticketing service, new e-tickets are issued instead.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

Successfully re-issued e-tickets.

502 Response

Examples

Response 200

Successfully issue or reissue new e-tickets:

json



```
{  
  "ticketNumbers": [  
    "0286010103190",  
    "0286010103191"  
  ]  
}
```

Description: Example response when issuance or re-issuance of e-tickets is successful.

GET /api/nsk/v1/booking/eTickets/validation

Summary: Checks if the booking in state is valid for e-ticketing operations.

Description

Request Details

Responses

200 Response

POST /api/nsk/v1/booking/fareOverride/journey/{journeyKey}

Summary: Replaces an existing fare price with an amount specified.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.

Request Body

The fare override request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v2/booking/fareRules

Summary: Retrieves the collection of all fare rules from the current booking in state.

Description

Note that this defaults to returning the fare rules in html format. The html format may not be entirely culture sensitive, so setting the convertToHtml flag to false will allow users some flexibility when converting the content and the fare rules will be returned as an rtf string.

Request Details

Parameters

Name	Location	Type	Required	Description
convertToHtml	query	boolean	No	Whether to convert to html. Defaults to true.

Responses

200 Response

The collection of fare rules.

404 Response

POST

/api/nsk/v2/booking/fareRules/category50/journeys/{journeyKey}/segments/{segmentKey}

Summary: Get the category 50 fare rule information for a specific segment on a booking.

Description

This is functionally a GET but POST is used to reduce URL length overhead.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
segmentKey	path	string	Yes	The segment key.
cultureCode	query	string	No	the culture code.
convertToHtml	query	boolean	No	If true, converts the response's Text value from RTF to HTML. Defaults to true.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/fareRules/fare/{fareKey}

Summary: Retrieves a fare rule from the current booking in state based on the specific fare key.

Description

Note that this defaults to returning the fare rules in html format. The html format may not be entirely culture sensitive, so setting the convertToHtml flag to false will allow users some flexibility when converting the content and the fare rules will be returned as an rtf string.

Request Details

Parameters

Name	Location	Type	Required	Description
fareKey	path	string	Yes	The unique fare key.
convertToHtml	query	boolean	No	Whether to convert to html. Defaults to true.

Responses

200 Response

The fare rule for the specific fare rule key.

404 Response

GET /api/nsk/v2/booking/fareRules/journey/{journeyKey}

Summary: Retrieves the collection of all fare rules from the current booking in state for the specific journey key.

Description

Note that this defaults to returning the fare rules in html format. The html format may not be entirely culture sensitive, so setting the convertToHtml flag to false will allow users some flexibility when converting the content and the fare rules will be returned as an rtf string.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique journey key.
convertToHtml	query	boolean	No	Whether to convert to html. Defaults to true.

Responses

200 Response

The collection of fare rules for the specific journey.

404 Response

GET /api/nsk/v2/booking/fareRules/segment/{segmentKey}

Summary: Retrieves the collection of all fare rules from the current booking in state for the specific segment key.

Description

Note that this defaults to returning the fare rules in html format. The html format may not be entirely culture sensitive, so setting the convertToHtml flag to false will allow users some flexibility when converting the content and the fare rules will be returned as an rtf string.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The unique segment key.
convertToHtml	query	boolean	No	Whether to convert to html. Defaults to true.

Responses

200 Response

The collection of fare rules for the specific segment.

404 Response

GET /api/nsk/v1/booking/fee

Summary: Gets the fee price.

Description

Only service and penalty fee types support a price that is configured by the booking in state. Depending on the configuration, the passenger key will be needed. For other fee types the price amount is manually provided.

Request Details

Parameters

Name	Location	Type	Required	Description
PassengerKey	query	string	No	The passenger key.

```
plaintext 
```

| Origin | query | string | No | The origin station code. || CollectedCurrencyCode | query | string | No | The collected currency code. || FeeCode | query | string | No | The fee code. || FeePricingMode | query | string | No | Determines if fees are to be paid in points or with currency.

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly |

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/fee

Summary: Manually adds a new fee.

Description

Penalty fees, spoilage fees, and service fees can be created manually. Other types are automatically created by the system. Depending on the configuration, the passenger key will be needed. If a passenger key was added to the request, the API will attempt to provide a location header url to point to the newly created fee using the passenger key and the passenger fee key from the new fee. However, depending on the fee type, this url may not always be valid. Spoilage fees for example are applied to the primary passenger, so if a non-primary passenger key was added on the request, the resulting url constructed by the API may result in a 404 - Not Found response.

```
plaintext 
```

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The fee request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT /api/nsk/v1/booking/fee/{feeKey}

Summary: Overrides a fee amount.

Description

When the fee override amount is zero, New Skies keeps or removes passenger fees with respect to the Store Confirming Waived Fees system setting. This endpoint returns a HTTP 200 on either case. Please retrieve the state booking afterwards to verify the absence or presence of the

overridden fee.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
feeKey	path	string	Yes	The fee key.

Request Body

The fee request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

DELETE /api/nsk/v1/booking/fee/{feeKey}

Summary: Deletes a fee.

Description

If a fee is deleted during the same commit in which it was added, then the fee will not be present on the booking -- It is truly deleted If a fee is waived after it has been committed to a booking, then the fee will still be present, along with a "FEE OR" fee indicating the override is 0. This behavior can be changed by checking the "Store Confirming Waived Fees" box in the Booking Settings dialog in utilities (SYS Users Only). If this box is checked, then all fees will behave in the second manner shown above -- meaning that an uncommitted fee that is overridden to 0 will not be removed from the booking. Additionally, if the fee was added pre-commit and the SkySpeed setting 'Allow Fees to be Deleted' is enabled for that user, then they will be able to delete the fee without needing waive fee permissions. Post-commit we still require waive fee permissions to override a fee.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
feeKey	path	string	Yes	The fee key request.

Responses

200 Response

PUT /api/nsk/v1/booking/fees

Summary: Overrides multiple fee amounts.

Description

When the fee override amount is zero, New Skies keeps or removes passenger fees with respect to the Store Confirming Waived Fees system setting. This endpoint returns a HTTP 200 on either case. Please retrieve the state booking afterwards to verify the absence or presence of the overridden fees.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The multiple fee override request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/nsk/v1/booking/fees/{feeKey}/quote

Summary: Gets the override fee quote of an existing fee.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
feeKey	path	string	Yes	The fee key of an existing fee.
Amount	query	number (decimal)	No	The override amount.

plaintext



| CollectedCurrencyCode | query | string | No | The collected currency code. |

Responses

200 Response

400 Response

404 Response

Examples

Response 200

Retrieve Override Fee Quote Response:

```
json

{
  "data": {
    "feeOverrideCharges": [
      {
        "amount": 100.0,
        "code": "NFSF",
        "detail": "Non Flight",
        "type": 6,
        "collectType": 0,
        "currencyCode": "USD",
        "foreignCurrencyCode": "USD",
        "foreignAmount": 100.0,
        "ticketCode": "NSF"
      }
    ]
  }
}
```

Description: Example response for successfully retrieving the override fee quote response.

Response 400

With No Booking in State:

```
json

{
  "errors": [
    {
      "id": "505b55d3-cac8-93a9-4d25-131256cdd551",
      "code": "nsk:Booking:NoBookingInState",
      "message": "nsk:Booking:NoBookingInState",
      "type": "Validation",
      "details": null,
      "rawMessage": "No booking was found in session, cannot perform operation.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if there is no booking in state.

With Invalid Passenger Key:

```
json

{
  "errors": [
    {
      "id": "2987bed7-5690-5050-55e9-819b8551b344",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'PassengerKey' with value 'MCFBASDFRFQ-' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if passenger key is invalid.

With Invalid Currency Code:

```
json
{
  "errors": [
    {
      "id": "10daf3f4-82c2-cf99-21ba-169fadae0b10",
      "code": "nsk:Validation:UnknownSellCurrencyCode",
      "message": "nsk:Validation:UnknownSellCurrencyCode",
      "type": "Validation",
      "details": {
        "currencyCode": "SDK"
      },
      "rawMessage": "The currency code SDK does not exist or is not active.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the currency code is invalid.

With Invalid Fee Key:

```
json
{
  "errors": [
    {
      "id": "ea9e908c-8979-85d8-c9ed-886014d6fc46",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'FeeKey' with value 'MT21XF' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if fee key is invalid.

With Invalid Fee Code:

```
json
{
  "errors": [
    {
      "id": "bf7c1a48-a4bf-ddf7-2413-93959ba6a3b3",
      "code": "nsk-server:FeeCalculation",
      "message": "nsk-server:FeeCalculation",
      "type": "Validation",
      "details": null,
      "rawMessage": "Cannot quote overrides of type NameChangeFee for new fees.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee code is not allowed to be quoted.

With Null Fee Code:

```
json
{
  "errors": [
    {
      "id": "6b34775e-d264-f3a6-4034-62daf6699617",
      "code": "GetNewOverrideFeeQuoteRequestBase.FeeCode:RequiredAttribute",
      "message": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "GetNewOverrideFeeQuoteRequestBase",
        "member": "FeeCode"
      },
      "rawMessage": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee code is null

With Null Fee Amount:

```
json
{
  "errors": [
    {
      "id": "6b34775e-d264-f3a6-4034-62daf6699617",
      "code": "GetNewOverrideFeeQuoteRequestBase.FeeCode:RequiredAttribute",
      "message": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "GetNewOverrideFeeQuoteRequestBase",
        "member": "Amount"
      },
      "rawMessage": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee amount is null

With Allow Negative Fees Role Setting Off:

```
json
{
  "errors": [
    {
      "id": "a49a0074-9b6d-9e30-83c3-1c21a56d748c",
      "code": "nsk:Validation:RoleSettingsProhibit",
      "message": "nsk:Validation:RoleSettingsProhibit",
      "type": "Validation",
      "details": null,
      "rawMessage": "Role settings prohibit a negative fee.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee amount is negative or the fee type is not allowed to override, and the role setting is turned off.

With Inactive Fees:

```
json   
{  
  "errors": [  
    {  
      "id": "bee189bc-edfc-263b-edc1-33cbf8b709cf",  
      "code": "nsk:Fee:NotFound",  
      "message": "nsk:Fee:NotFound",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "The fee was not found.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example Error if the agent attempted to sell inactive fees.

With Unauthorized Overriding of Service Fees:

```
json   
{  
  "errors": [  
    {  
      "id": "7911302f-46af-3dc8-5714-b2823fec853d",  
      "code": "nsk-server:AuthorizationSSRNotAllowed",  
      "message": "Agent is not authorized to override fees.",  
      "type": "Validation",  
      "details": {  
        "message": "Agent is not authorized to override fees."  
      },  
      "rawMessage": "Agent is not authorized to override fees.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example Error if the agent is not authorized to override a service fee.

With Unauthorized Selling of Service Fees:

```
json   
{  
  "errors": [  
    {  
      "id": "cfdacd38-5e2c-ab09-9fce-86722e945f0b",  
      "code": "nsk-server:AuthorizationServiceFeeNotAllowed",  
      "message": "Agent is not authorized to sell service fee FLO.",  
      "type": "Validation",  
      "details": {  
        "message": "Agent is not authorized to sell service fee FLO."  
      },  
      "rawMessage": "Agent is not authorized to sell service fee FLO.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example Error if the agent is not authorized to sell a service fee.

DELETE /api/nsk/v1/booking/fees/penalty

Summary: Waives the penalty fee of all journeys in the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

DELETE /api/nsk/v1/booking/fees/spoilage

Summary: Waives the spoilage fee of all journeys in the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

DELETE /api/nsk/v1/booking/groupName

Summary: Deletes the group name of the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

PUT /api/nsk/v2/booking/groupName

Summary: Changes the group name of the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The booking group name request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

GET /api/nsk/v2/booking/hold/available

Summary: Retrieves the recommended hold date if hold is available.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/booking/indiaGstRegistrationRequirement

Summary: Returns true if it is expected to prompt the user for India GST registration details.

Description

Request Details

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/itinerary/print

Summary: Sends the booking itinerary notification to the specified printer. Automatically commits uncommitted changes to the state booking before queuing the booking to a print notification queue.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The print itinerary request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

202 Response

400 Response

GET /api/nsk/v1/booking/penalties

Summary: Gets the booking penalty from the booking in state.

Description

Request Details

Responses

200 Response

404 Response

DELETE /api/nsk/v1/booking/penalties/{cancelledJourneyKey}

Summary: Waives the penalty fees of the specified cancelled journey.

Description

If the journey key specified is for a non-cancelled journey, i.e. the journey exists in the booking in state, an exception will be thrown. To waive the penalty fee of a journey in the booking in state, please use /api/nsk/v1/booking/journeys/{journeyKey}/fees/penalty instead.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
cancelledJourneyKey	path	string	Yes	The journey key of the cancelled journey.

Responses

200 Response

PUT /api/nsk/v1/booking/pointOfSale

Summary: Updates the booking point of sale.

Description

The ISO country code cannot be updated for a committed booking. This value can only be updated on an uncommitted booking.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Details to update the point of sale.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/booking/pointOfSale

Summary: Updates the booking point of sale.

Description

The ISO country code cannot be updated for a committed booking. This value can only be updated on an uncommitted booking.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Details to update the point of sale.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

POST /api/nsk/v1/booking/prepaidExternal

Summary: Creates a new prepaid external payment on the booking in state.

Description

This endpoint requires agent permissions.

plaintext



```
The payment fields are dynamic to the criteria of your payment provider. Here is an example request for a credit card payment that does not require address, zip, etc.  
{ "amount": 10, "currencyCode": "USD", "paymentFields": {"ACCTNO":"411111111111", "EXPDATE":"10/8/2017", "CC::AccountHolderName":"Bob Smith", "CC::VerificationCode":"111"}}
```

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
prepaidApprovalCode	query	string	No	The prepaid approval code. This maps to a payment field with PrepaidApprovalCode as the system field type
(configured in Utilities - Payment Types).				

plaintext



Request Body

The payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The booking payments response.

PUT /api/nsk/v1/booking/price

Summary: Applies fares, discounts and taxes to the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The price request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/price/validation

Summary: Validates the fare prices on the booking in state.

Description

This endpoint is only recommended to be used with Fare Manager Plus and CAT1 validation is not fully supported in this endpoint outside of 4.5.2.

Request Details

Responses

200 Response

400 Response

Examples

Response 400

Failed Fare Restrictions:

```
json   
{  
  "errors": [  
    {  
      "id": "26b205f5-6db8-4b6f-ab9b-5697d5f9e3f7",  
      "code": "nsk:Booking:FailedFareRestriction",  
      "message": "nsk:Booking:FailedFareRestriction",  
      "type": "Error",  
      "details": {  
        "failedFareRestrictions": "Failed passenger eligibility restrictions\nCarrier: XY  Rule Number: DA0X  Fare Basis: HBUDDY",  
        "rawMessage": "Failed passenger eligibility restrictions\nCarrier: XY  Rule Number: DA0X  Fare Basis: HBUDDY  New Skies Defa",  
        "exception": null,  
        "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r",  
      },  
      "data": null  
    }  
  ]  
}
```

Description: Example response when booking has failed fare restrictions

POST /api/nsk/v1/booking/promotion

Summary: Sets a promotion code to a booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Promotion request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

Bad request.

409 Response

Concurrent changes were being made in the same session.

Examples

Request

Promotion request:

```
json   
{  
  "promotionCode": "PROMO1",  
  "organizationCode": null  
}
```

Description: Example request for adding a promotion code

Promotion request with organization code:

```
json   
{  
  "promotionCode": "PROMO1",  
  "organizationCode": "Organization ABC"  
}
```

Description: Example request for adding a promotion code

Response 400

Promotion Already On Booking Error:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Validation:ObjectAlreadyExists",  
      "message": "nsk:Validation:ObjectAlreadyExists",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "Object already exists.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ],  
  "data": null  
}
```

Description: Example error when attempting to add a promotion to a booking that already has a promotion applied.

Requested Promotion Code Too Long:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "PromotionRequest.PromotionCode:StringLengthAttribute",
      "message": "The field PromotionRequest must be a string with a maximum length of 8.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The field PromotionRequest must be a string with a maximum length of 8.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to add a promotion code with more than 8 characters.

Non-Existent Promotion Code:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:PromotionNotFound",
      "message": "nsk-server:PromotionNotFound",
      "type": "Information",
      "details": null,
      "rawMessage": "Specified promotion does not exist: PROMO22",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to use a promotion code that does not exist.

PUT /api/nsk/v1/booking/promotion

Summary: Updates a promotion code to a booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Promotion request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

400 Response

Bad request.

409 Response

Concurrent changes were being made in the same session.

Examples

Request

Promotion request:

```
json   
{  
  "promotionCode": "PROMO1",  
  "organizationCode": null  
}
```

Description: Example request for adding a promotion code

Promotion request with organization code:

```
json   
{  
  "promotionCode": "PROMO1",  
  "organizationCode": "Organization ABC"  
}
```

Description: Example request for adding a promotion code

Response 200

Promotion Updated:

```
json   
{  
  "data": null  
}
```

Description: Example when the promotion has been successfully updated

Expired Promotion:

```
json   
{  
  "messages": [  
    {  
      "rawValue": "Promotion AUTPCAA is not valid for organization: ORG1.",  
      "code": "PromotionNotApplied",  
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",  
      "type": "Information",  
      "value": "PromotionNotApplied",  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": null  
}
```

Description: Example warning when attempting to add an expired promotion

Response 400

Same Promotion Already On Booking Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:NotModified",
      "message": "nsk:Booking:NotModified",
      "type": "Validation",
      "details": null,
      "rawMessage": "Booking was not modified.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to update the promotion on the booking to the same promotion that is already on the booking.

Requested Promotion Code Too Long:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "PromotionRequest.PromotionCode:StringLengthAttribute",
      "message": "The field PromotionRequest must be a string with a maximum length of 8.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The field PromotionRequest must be a string with a maximum length of 8.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to add a promotion code with more than 8 characters.

Non-Existent Promotion Code:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:PromotionNotFound",
      "message": "nsk-server:PromotionNotFound",
      "type": "Information",
      "details": null,
      "rawMessage": "Specified promotion does not exist: PROMO22",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to use a promotion code that does not exist.

No Promotion Code On Booking:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Validation:ObjectNotFound",
      "message": "nsk:Validation:ObjectNotFound",
      "type": "Validation",
      "details": null,
      "rawMessage": "The object is not found.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to update a promotion code when none is on the booking.

DELETE /api/nsk/v1/booking/promotion

Summary: Deletes the promotion code on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

OK. Request completed successfully.

400 Response

Bad request.

409 Response

Concurrent changes were being made in the same session.

Examples

Response 400

No Promotion Code On Booking:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Validation:ObjectNotFound",
      "message": "nsk:Validation:ObjectNotFound",
      "type": "Validation",
      "details": null,
      "rawMessage": "The object is not found.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when attempting to delete a promotion code when none is on the booking.

POST /api/nsk/v3/booking/queue

Summary: Adds the booking from state to a booking queue.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The booking request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

400 Response

Examples

Response 400

Enqueue Booking, Queue Code Does Not Exist:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingQueueDoesNotExist",
      "message": "nsk-server:BookingQueueDoesNotExist",
      "type": "Information",
      "details": null,
      "rawMessage": "Booking Queue 'ZZZZZ' does not exist in the DB..",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when trying to add a booking to a queue, and the requested queue code does not exist.

Manual Queueing for Queue Code Not Allowed:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:InvalidSelfServiceRebookingQueueing",
      "message": "nsk-server:InvalidSelfServiceRebookingQueueing",
      "type": "Information",
      "details": null,
      "rawMessage": "Manual queuing from a Self-Service Rebooking Booking Queue is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when trying to add a booking to a queue, and manual queueing from that queue is not allowed.

DELETE /api/nsk/v2/booking/queue

Summary: Removes the booking in state from a booking queue.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The booking request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

DELETE /api/nsk/v1/booking/queues/{bookingQueueKey}

Summary: Removes the booking in state from the specified booking queue item.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueKey	path	string	Yes	The unique key of the booking queue to dequeue from.

Request Body

The request for the dequeue operation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

GET /api/nsk/v1/booking/recordLocators

Summary: Gets all of the record locators on the in-state booking.

Description

Request Details

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/recordLocators

Summary: Adds a third party record locator to the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The record locator request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

DELETE /api/nsk/v1/booking/recordLocators

Summary: Deletes all record locators from the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

GET /api/nsk/v1/booking/recordLocators/{recordLocatorKey}

Summary: Gets the record locator associated with the record locator key for the in-state booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocatorKey	path	string	Yes	No description

Responses

200 Response

The record locator.

404 Response

PUT /api/nsk/v1/booking/recordLocators/{recordLocatorKey}

Summary: Replaces the data of a third party record locator on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocatorKey	path	string	Yes	The record locator key to update.

Request Body

The record locator update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/booking/recordLocators/{recordLocatorKey}

Summary: Patches the data of a third party record locator on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocatorKey	path	string	Yes	The record locator key to update.

Request Body

The record locator update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/booking/recordLocators/{recordLocatorKey}

Summary: Deletes a record locator from the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocatorKey	path	string	Yes	The record locator key.

Responses

200 Response

PUT /api/nsk/v1/booking/reinstate

Summary: Reinstates a booking with a status of hold canceled.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

400 Response

404 Response

500 Response

Examples

Response 200

Reinstate Successful:

```
json   
{  
  "data": null  
}
```

Description: Example when booking is Reinstate Successfully

Failed To Assign All Seats Message:

```
json   
{  
  "messages": [  
    {  
      "rawValue": "Failed to assign seat(s) for passenger Aurelio David on the flight NV5682 SLCBOS Detail: Seats cannot be assigned",  
      "code": "FailedToAssignAllSeats",  
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are for informational purposes only.",  
      "type": "FailedToAssignAllSeats",  
      "value": null,  
      "status": 0,  
      "details": {  
        "reinstateGeneralMessage": "Failed to assign seat(s) for passenger Aurelio David on the flight NV5682 SLCBOS Detail: Seats cannot be assigned"  
      }  
    }  
  ],  
  "data": null  
}
```

Description: Example OSI Message when reinstating a booking with no available seats

SSR Is Unavailable Message:

```

json

{
  "messages": [
    {
      "rawValue": "A SSR could not be sold for the following SSR, passenger and flight: LEGX|MCFBRFQ-|NV5683 |DEN|SLC|2023-11-17T0
      "code": "SSRIsUnavailable",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are
      "type": "SSRIsUnavailable",
      "value": null,
      "status": 0,
      "details": {
        "ssrCode": "LEGX",
        "passengerKey": "MCFBRFQ-",
        "carrierCode": "NV",
        "flightNumber": "5683",
        "opSuffix": null,
        "origin": "DEN",
        "destination": "SLC",
        "departureDate": "2023-11-17T00:00:00"
      }
    }
  ],
  "data": null
}

```

Description: Example OSI Message when reinstating a booking with no available SSRs

Failed To Reinstate Journey Message:

```

json

{
  "messages": [
    {
      "rawValue": "Failed to reinstate journey: 20231101 NV 5682 SLCBOS. Reason: The requested class of service is sold out.",
      "code": "FailedToReinstateJourney",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are
      "type": "FailedToReinstateJourney",
      "value": null,
      "status": 1,
      "details": {
        "reinstateWarningMessage": "Failed to reinstate journey: 20231101 NV 5682 SLCBOS. Reason: The requested class of service
      }
    }
  ],
  "data": null
}

```

Description: Example OSI Message when reinstating a multi journey booking where there is at least one journey with no available AUs and at least one journey that succeeded to reinstate.

Response 400

Reinstate Restricted On Non HoldCanceled Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:ReinstateRestrictedOnNonHoldCanceled",
      "message": "nsk:Booking:ReinstateRestrictedOnNonHoldCanceled",
      "type": "Validation",
      "details": null,
      "rawMessage": "Unable to reinstate a booking that is not hold canceled.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when Reinstating a booking that is not Hold Canceled.

No Booking In State Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:NoBookingInState",
      "message": "nsk:Booking:NoBookingInState",
      "type": "Validation",
      "details": null,
      "rawMessage": "No booking was found in session, cannot perform operation.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when Reinstating without a booking in state.

Failed To Reinstate Journey Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "FailedToReinstateJourney",
      "message": "FailedToReinstateJourney",
      "type": "Error",
      "details": {
        "reinstateErrorMessage": "Failed to reinstate journey: 20231101 NV 5682 SLCBOS. Reason: The requested class of service is not offered",
        "rawMessage": "Failed to reinstate journey: 20231101 NV 5682 SLCBOS. Reason: The requested class of service is not offered",
        "exception": null,
        "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
      }
    },
    "data": null
  }
}
```

Description: Example error when all journeys failed to reinstate.

DELETE /api/nsk/v1/booking/reset

Summary: Clears the current booking state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

Http response message.

PUT /api/nsk/v1/booking/salesChannel

Summary: Overrides the effective sales channel for the booking in state.

Description

For the overriden channel to be implemented, the booking in state must be committed after calling this endpoint.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
channelType	query	string	No	The sales channel type to override to.
Enumeration values: 0 = Direct, 1 = Web, 2 = Api, 3 = DigitalApi, 4 = DigitalWeb, 5 = Ndc				

Responses

200 Response

Http response message.

500 Response

GET /api/nsk/v2/booking/status

Summary: Gets the status of the booking commit and returns the booking.

Description

There are certain booking data that are not saved to state, such as payment attachments. This endpoint will return the booking data so that non persisted information is returned when and only when the status code is 200. This is the only time the data will be available.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

Ok, booking has processed.

202 Response

Accepted, booking is still processing.

PUT /api/nsk/v1/booking/systemCode

Summary: Updates the system code of the state booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The updated system code.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/booking/systemCode

Summary: Removes the system code of the state booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

500 Response

GET /api/nsk/v1/booking/validation

Summary: Evaluates the booking in state prior to commit or payment processing.

Description

The response indicates the forecasted status that will be applied to the booking during commit and if a held seat warning should be issued.

Request Details

Responses

200 Response

404 Response

POST

/api/nsk/v1/booking/verifiedTravelDocuments/segments/{segmentKey}/passengers/{passengerKey}

Summary: Adds a new verified travel document to a passenger or the associated infant.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext



The following items are required to use this endpoint.

Agent permissions.

Government Security Drivers need to be enabled under government security configurations.

Government Security Drivers need to be configured to approve the given document type.

The given document must be a document type approved by customs.

The target passenger must be either CheckedIn or Boarded on the target segment.

Due to the nature of this endpoint there is no data returned. To confirm successful update you can find an entry in the booking using GET `api/nsk/v1/booking` or view passenger segment information by pulling the booking using GET `/api/nsk/v1/bookings/{recordLocator}`

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The target segment key.
passengerKey	path	string	Yes	The target passenger key.

Request Body

The request for updating verified travel documents on a bookings.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Request

Example verified travel document:

json



```
{  
  "documentTypeCode": "P",  
  "issuedByCode": "US",  
  "isInfant": false  
}
```

Description: An example of a basic verified travel document.

Response 400

Invalid Record Locator Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "recordLocator": "9SkFNRegdQRYuQ3YfA2g9"
      },
      "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3YfA2g9' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "passengerKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when passenger key is invalid.

Invalid Segment Key Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "segmentKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'SegmentKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when segment key is invalid.

Invalid Lift Status Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:VerifiedTravelDocuments:InvalidLiftStatus",
      "message": "Updating verified travel documents requires the passenger to have a lift status of checked in or boarded. Lift s",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}
```

Description: Example error when lift status is invalid.

Invalid Document Type Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:VerifiedTravelDocuments:InvalidDocumentType",
      "message": "The document type code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}
```

Description: Example error when document type is invalid.

Invalid Issued by Code Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:VerifiedTravelDocuments:InvalidIssuedByCode",
      "message": "The issued by code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}
```

Description: Example error when document issued by code is invalid.

Booking/Addons Endpoints

GET /api/nsk/v1/booking/addons/{addOnKey}/payments

Summary: Gets a specific add-on's allowed payment methods.

Description

If the add on's payment required flag is set to true, this endpoint will return the allowed payment options for the add on. If set to false, null will be returned.

Request Details

Parameters

Name	Location	Type	Required	Description
addOnKey	path	string	Yes	The unique add on key.

Responses

200 Response

The add on's allowed payments.

404 Response

POST /api/nsk/v1/booking addons/{addOnKey}/payments

Summary: Adds a payment to a specific add-on on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
addOnKey	path	string	Yes	The unique add on key.

Request Body

The order payment to be added.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Status created.

GET /api/nsk/v1/booking addons/payments

Summary: Gets all the available add-ons allowed payment methods for all the add-ons on the booking in state.

Description

Request Details

Responses

200 Response

The dictionary of add on allowed payments.

404 Response

PUT /api/nsk/v1/booking addons sync

Summary: Syncs the prices of the Travel Commerce addons.

Description

This endpoint should be called prior to displaying the total booking cost and collecting payment for Travel Commerce items. It will trigger New Skies to query Travel Commerce for the items in the cart.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

500 Response

Booking/Boardingpasses Endpoints

POST /api/nsk/v3/booking/boardingpasses/journey/{journeyKey}

Summary: Retrieves the version of the boarding passes based on a specific journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey to retrieve boarding passes for.

Request Body

The passenger/segment filter request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The boarding passes.

404 Response

POST /api/nsk/v1/booking/boardingpasses/m2d/journey/{journeyKey}

Summary: Retrieves the specific journeys boarding passes using the M2D barcode type. This endpoint will only give a single barcode back even if the journey contains multiple segments. Note that if the journey has only one segment the M2D barcode will look just like the S2D barcode.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The specific journey to get boarding passes for.

Request Body

The boarding pass passenger filter request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The boarding passes for the journey.

404 Response

POST /api/nsk/v1/booking/boardingpasses/s2d/journey/{journeyKey}

Summary: Retrieves the specific journeys boarding passes for each segment using the S2D barcode type. This endpoint will provide a single barcode for each segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The specific journey to get boarding passes for.

Request Body

The boarding pass passenger filter request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The boarding passes for the journey.

404 Response

POST /api/nsk/v3/booking/boardingpasses/segment/{segmentKey}

Summary: Retrieves the version of the boarding passes based on a specific segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment to retrieve boarding passes for.

Request Body

The passenger filter request

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The boarding passes.

404 Response

Booking/Checkin Endpoints

DELETE /api/nsk/v1/booking/checkin/journey/{journeyKey}

Summary: Removes the checkin status of passengers for a specific journey in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v3/booking/checkin/journey/{journeyKey}

Summary: Checks in a collection of passengers to a specific journey in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

- When debug mode is disabled, warning and error messages will be masked if the `SystemOptions.HideErrorMessages` is set to true.
- When debug mode is disabled, warning and error messages will be returned to agents only if the `SystemOptions.HideErrorMessages` is set to false.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key of the segments to checkin.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 201

Check-in Successful:

```
json

{
  "data": null
}
```

Description: Example when check-in is successful.

Checkin Valid with displaying warning message to agents.:

```
json

{
  "messages": [
    {
      "code": "CheckinPassengerWarning",
      "type": "Warning",
      "value": "AUT TEST Authorization Response Free Text\nn8501:AU:OK TO BOARD\nDocuments are required for this flight.",
      "status": 0,
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_MDgvMjEvMjAyNCAwOTowMH5TWUR_MDgvMjEvMjAyNCAxMjowMH5_"
      }
    }
  ],
  "data": null
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This requires Digital API configuration to be set such that warning messages can be returned to agents only when debug is turned off.

Checkin Valid with a masked warning message.:

```
json

{
  "messages": [
    {
      "code": "CheckinPassengerWarning",
      "type": "Warning",
      "value": "CheckinPassengerWarning",
      "status": 0,
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_MDgvMjEvMjAyNCAwOTowMH5TWUR_MDgvMjEvMjAyNCAxMjowMH5_"
      }
    }
  ],
  "data": null
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This is what non-agents will see regardless of Digital API configuration for displaying warning messages when debug is off. This is also what an agent would see if the corresponding API configuration is not set to allow warning messages to be shown.

Response 400

Checkin Invalid:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:Invalid",
      "message": "nsk:Checkin:Invalid",
      "type": "Validation",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Checkin request is invalid. Please refer to checkin requirements for more details.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    ],
    "data": null
  }
}
```

Description: Example error when check in requirements are not met. The /requirements endpoint should be called before attempting a checkin.

Seat Missing:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1223",
      "message": "nsk:Checkin:TypeID:1223",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Baggins/Bilbo missing seat assignments on flight number 2452.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    ],
    "data": null
  }
}
```

Description: In this example the optional request parameter `seatsRequired` was set to true. One passenger had a seat assigned and one did not.

Inhibited:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3096",
      "message": "nsk:Checkin:TypeID:3096",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CAL-9444",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    },
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3088",
      "message": "nsk:Checkin:TypeID:3088",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CAL-9444",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}

```

Description: In this example the passenger meets all the requirements but is inhibited due to government security.

Missing verified document with agent visible error message:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1236",
      "message": "Baggins/Bilbo missing verified documents.",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered."
    }
  ],
  "data": null
}

```

Description: This example shows both the expected response when a passenger is missing verified documents, as well as what a logged-in agent would see if the API configuration is set correctly. This is an example response when debug mode is turned off.

GET /api/nsk/v2/booking/checkin/journey/{journeyKey}/requirements

Summary: Gets the checkin pre-validation requirements for a specific journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key to be pre validate checkin with.

Responses

200 Response

The checkin validation requirements.

404 Response

Examples

Response 200

Valid Response No Restrictions:

json



```
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": {
          "documentsOnFile": [
            "P",
            "V"
          ],
          "governmentDocumentInstances": [
            {
              "governmentInstance": "USDHS",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                },
                {
                  "group": 5,
                  "documentTypeCode": "M"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            },
            {
              "governmentInstance": "CBIA",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            }
          ],
          "isValid": true
        },
        "addressRequirements": {
          "isValid": false,
          "addressesOnFile": [
            "Address1"
          ],
          "unhandledAddresses": [
            "Address2"
          ],
          "invalidSsrs": null,
          "requiredSsrs": null,
        }
      }
    }
  }
}
```

```

        "restrictions": null,
        "isValid": true
    },
    "governmentProgramRequirements": {
        "usdhs": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "USDHS",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        },
        "cbsa": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "CBSA",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        }
    },
    "restrictions": null,
    "isValid": true
}
}

```

Description: Example Response

Response with Restrictions:

json

```
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": null,
        "addressRequirements": null,
        "invalidSsrs": null,
        "requiredSsrs": null,
        "restrictions": [
          {
            "restriction": 12,
            "description": "Missing Documents"
          },
          {
            "restriction": 1,
            "description": "Missing Destination Address"
          }
        ],
        "isValid": false
      }
    },
    "governmentProgramRequirements": {
      "usdhs": {
        "governmentInstance": null,
        "requiredDocumentFields": null,
        "additionalDataRequired": null,
        "programDescription": "USDHS",
        "messageType": null,
        "minVerifiedDocumentCount": 1
      }
    },
    "restrictions": [
      4
    ],
    "isValid": false
  }
}
```

Description: Example Response

GET /api/nsk/v1/booking/checkin/journey/{journeyKey}/status

Summary: Gets the passengers lift status for a specific journey based on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey to check lift statuses.

Responses

200 Response

The collection of passenger lift statuses for the journey.

404 Response

DELETE /api/nsk/v1/booking/checkin/segment/{segmentKey}

Summary: Removes the checkin status of passengers for a specific journey's segment in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v3/booking/checkin/segment/{segmentKey}

Summary: Checks in a collection of passengers to a specific journey's segment in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

- When debug mode is disabled, warning and error messages will be masked if the `SystemOptions.HideErrorMessages` is set to true.
- When debug mode is disabled, warning and error messages will be returned to agents only if the `SystemOptions.HideErrorMessages` is set to false.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key of the segment to checkin.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 201

Check-in Successful:

```
json
{
  "data": null
}
```

Description: Example when check-in is successful.

Checkin Valid with displaying warning message to agents.:

```
json
{
  "messages": [
    {
      "code": "CheckinPassengerWarning",
      "type": "Warning",
      "value": "AUT TEST Authorization Response Free Text\nn8501:AU:OK TO BOARD\nDocuments are required for this flight.",
      "status": 0,
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_mDgvMjEvMjAyNCAwOTowMH5TWUR_mDgvMjEvMjAyNCAxMjowMH5_"
      }
    }
  ],
  "data": null
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This requires Digital API configuration to be set such that warning messages can be returned to agents only when debug is turned off.

Checkin Valid with a masked warning message.:

```
json
```

```
{  
  "messages": [  
    {  
      "code": "CheckinPassengerWarning",  
      "type": "Warning",  
      "value": "CheckinPassengerWarning",  
      "status": 0,  
      "details": {  
        "passengerKey": "MCFBRFQ-",  
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_mDgvMjEvMjAyNCAwOTowMH5TWUR_mDgvMjEvMjAyNCAxMjowMH5_"  
      }  
    },  
    "data": null  
  ]  
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This is what non-agents will see regardless of Digital API configuration for displaying warning messages when debug is off. This is also what an agent would see if the corresponding API configuration is not set to allow warning messages to be shown.

Response 400

Checkin Invalid:

```
json
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Checkin:Invalid",  
      "message": "nsk:Checkin:Invalid",  
      "type": "Validation",  
      "details": {  
        "passengerKey": "MCFBRFQ-",  
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_mDkvMDYvMjAyNCAwOTowMH5NTkx_mDkvMDcvMjAyNCAwNTowMH5_"  
      },  
      "rawMessage": "Checkin request is invalid. Please refer to checkin requirements for more details.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
    "data": null  
  ]  
}
```

Description: Example error when check in requirements are not met. The /requirements endpoint should be called before attempting a checkin.

Seat Missing:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1223",
      "message": "nsk:Checkin:TypeID:1223",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Baggins/Bilbo missing seat assignments on flight number 2452.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "data": null
    }
  ]
}
```

Description: In this example the optional request parameter `seatsRequired` was set to true. One passenger had a seat assigned and one did not.

Inhibited:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3096",
      "message": "nsk:Checkin:TypeID:3096",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3088",
      "message": "nsk:Checkin:TypeID:3088",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "data": null
    }
  ]
}
```

Description: In this example the passenger meets all the requirements but is inhibited due to government security.

Missing verified document with agent visible error message:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1236",
      "message": "Baggins/Bilbo missing verified documents.",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: This example shows both the expected response when a passenger is missing verified documents, as well as what a logged-in agent would see if the API configuration is set correctly. This is an example response when debug mode is turned off.

GET /api/nsk/v2/booking/checkin/segment/{segmentKey}/requirements

Summary: Gets the checkin pre-validation requirements for a specific journey's segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key to pre validate checkin with.

Responses

200 Response

The checkin validation requirements.

404 Response

Examples

Response 200

Valid Response No Restrictions:

json



```
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": {
          "documentsOnFile": [
            "P",
            "V"
          ],
          "governmentDocumentInstances": [
            {
              "governmentInstance": "USDHS",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                },
                {
                  "group": 5,
                  "documentTypeCode": "M"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            },
            {
              "governmentInstance": "CBIA",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            }
          ],
          "isValid": true
        },
        "addressRequirements": {
          "isValid": false,
          "addressesOnFile": [
            "Address1"
          ],
          "unhandledAddresses": [
            "Address2"
          ],
          "invalidSsrs": null,
          "requiredSsrs": null,
        }
      }
    }
  }
}
```

```

        "restrictions": null,
        "isValid": true
    },
    "governmentProgramRequirements": {
        "usdhs": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "USDHS",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        },
        "cbsa": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "CBSA",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        }
    },
    "restrictions": null,
    "isValid": true
}
}

```

Description: Example Response

Response with Restrictions:

```

json
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": null,
        "addressRequirements": null,
        "invalidSsrs": null,
        "requiredSsrs": null,
        "restrictions": [
          {
            "restriction": 12,
            "description": "Missing Documents"
          },
          {
            "restriction": 1,
            "description": "Missing Destination Address"
          }
        ],
        "isValid": false
      }
    },
    "governmentProgramRequirements": {
      "usdhs": {
        "governmentInstance": null,
        "requiredDocumentFields": null,
        "additionalDataRequired": null,
        "programDescription": "USDHS",
        "messageType": null,
        "minVerifiedDocumentCount": 1
      }
    },
    "restrictions": [
      4
    ],
    "isValid": false
  }
}

```

Description: Example Response

GET /api/nsk/v1/booking/checkin/segment/{segmentKey}/status

Summary: Gets the passenger lift status for a specific segment based on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key in reference.

Responses

200 Response

The collection of passenger lift statuses for the journey.

404 Response

Booking/Contacts Endpoints

GET /api/nsk/v1/booking/contacts

Summary: Gets all the contacts on the booking.

Description

No description available

Request Details

Responses

200 Response

The collection of contacts.

404 Response

POST /api/nsk/v1/booking/contacts

Summary: Creates a new contact on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The new contact.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

409 Response

Concurrent changes were being made in the same session.

Examples

Response 201

Created contact:

```
json   
{  
  "data": null  
}
```

Description: A contact was created and added to the booking.

Response 400

With Invalid Contact Type Code:

```
json   
{  
  "errors": [  
    {  
      "id": "12bd2c3d-ddd4-5844-3sz5-3cg52c2e13n1",  
      "code": "nsk>Contact:InvalidContactTypeCode",  
      "message": "The contact type code 'S' is invalid.",  
      "type": "Validation",  
      "details": {  
        "contactTypeCode": "S"  
      },  
      "rawMessage": "The contact type code 'S' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The contact type code is invalid or not configured in the system. The list of valid contact type codes can be retrieved using the GET api/nsk/v1/resources/ContactTypes endpoint.

With Duplicate Contact Type Code:

```
json   
{  
  "errors": [  
    {  
      "id": "99cd1b2b-ccc3-4733-7fa4-2bf49b1d98c0",  
      "code": "nsk>Contact:DuplicateTypeCode",  
      "message": "Duplicate contact type 'P' detected.",  
      "type": "Validation",  
      "details": {  
        "contactTypeCode": "P"  
      },  
      "rawMessage": "Duplicate contact type 'P' detected.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The contact type code already exists in the booking.

With No Contact Type Code:

```
json

{
  "errors": [
    {
      "id": "5f490eab-3a4d-a783-14ed-25456aecf82f",
      "code": "Contact.ContactTypeCode:RequiredAttribute",
      "message": "The Contact field is required.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The Contact field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The contact type code in the request is not populated.

With Incomplete Booking Contact Info:

```
json

{
  "errors": [
    {
      "id": "666e7592-c107-3812-46e6-f7147dd3c5cd",
      "code": "nsk-server:BookingContactInfoIncomplete",
      "message": "Missing or invalid GST Registration Number.",
      "type": "Validation",
      "details": null,
      "rawMessage": "Missing or invalid GST Registration Number.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The booking contact info provided is incomplete.

With No Booking in State:

```
json

{
  "errors": [
    {
      "id": "505b55d3-cac8-93a9-4d25-131256cdd551",
      "code": "nsk:Booking:NoBookingInState",
      "message": "No booking was found in session, cannot perform operation.",
      "type": "Validation",
      "details": null,
      "rawMessage": "No booking was found in session, cannot perform operation.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: No booking was put into state before creating a contact.

With Null Request:

```

json
{
  "errors": [
    {
      "id": "292fcb67-e1de-eb62-3d1a-123a586b03a9",
      "code": "core:Validation:NullRequest",
      "message": "The request is null or couldn't be converted to the expected data type.",
      "type": "Error",
      "details": null,
      "rawMessage": "The request is null or couldn't be converted to the expected data type.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: The contact request field is null

GET /api/nsk/v1/booking/contacts/{contactTypeCode}

Summary: Gets a specific contact on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.

Responses

200 Response

The contact.

404 Response

PUT /api/nsk/v1/booking/contacts/{contactTypeCode}

Summary: Updates a specific contact on the booking based on type code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique type code. Please note that this should be a char and not a string.

Request Body

The modified contact.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/booking/contacts/{contactTypeCode}

Summary: Patches a specific contact on the booking based on type code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique type code. Please note that this should be a char and not a string.

Request Body

The patched contact request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/booking/contacts/{contactTypeCode}

Summary: Deletes a specific contact on the booking based on type code.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique type code. Please note that this should be a char and not a string.

Responses

200 Response

Ok.

GET /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers

Summary: Gets the collection of phone number for a specific contact on the booking.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.

Responses

200 Response

The collection of contact phone numbers.

404 Response

POST /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers

Summary: Creates a new phone number on a specific contact on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.

Request Body

The new phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType}

Summary: Gets the specific phone number from a specific contact on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.
phoneNumberType	path	string	Yes	The phone number type.
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Responses

200 Response

The phone number.

404 Response

PUT /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType}

Summary: Updates a specific phone number on a specific contact on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.
phoneNumberType	path	string	Yes	The phone number type.
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Request Body

The modified phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Created.

500 Response

DELETE /api/nsk/v1/booking/contacts/{contactTypeCode}/phoneNumbers/{phoneNumberType}

Summary: Deletes a specific phone number on a specific contact on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.
phoneNumberType	path	string	Yes	The phone number type.
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Responses

200 Response

The task.

GET /api/nsk/v1/booking/contacts/primary

Summary: Gets the primary contact based on configuration.

Description

Request Details

Responses

200 Response

The primary contact.

404 Response

POST /api/nsk/v1/booking/contacts/primary

Summary: Creates a new primary contact based on configuration.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The new contact.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

PUT /api/nsk/v1/booking/contacts/primary

Summary: Updates the primary contact.

Description

Request Details

Request Body

The modified contact request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/booking/contacts/primary

Summary: Patches the primary contact.

Description

Request Details

Request Body

The patched contact request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/booking/contacts/primary

Summary: Delete the primary contact.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

Ok.

Booking/History Endpoints

GET /api/nsk/v1/booking/history

Summary: Gets the current booking history in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Event	query	string	No	The event that triggered the history entry.

Enumeration values: 0 = Unknown, 1 = ConvertedHistory, 2 = FlightTimeChange, 3 = FlightDesignatorChange, 4 = AssignedSeat, 5 = RemoveSeat, 6 = AddedFlight, 7 = DeletedFlight, 8 = DeletedPassenger, 9 = NameChange, 10 = GroupNameChange, 11 = CancelledTicketing, 12 = ScheduleChange, 13 = AddedPayment, 14 = ServiceFee, 15 = QueuedPnr, 16 = UnqueuedPnr, 17 = DeletedComment, 18 = Divided, 19 = CheckedIn, 20 = CheckedOut, 21 = FareOverride, 22 = AddedBaggage, 23 = ChangedBaggageWeight, 24 = CheckedBaggage, 25 = RemovedBaggage, 26 = BoardedPassenger, 27 = UnboardedPassenger, 28 = ManualAuthorization, 29 = ManualDecline, 30 = UndoCancel, 31 = ItinerarySent, 32 = ContactChange, 33 = SsrAdded, 34 = FlightMoved, 35 = VerifiedDocument, 36 = RemovedVerifiedDocument, 37 = Promotion, 38 = BookingComment, 39 = CancelledSchedule, 40 = CancelServiceFee, 41 = OverrideServiceFee, 42 = AddedRecordLocator, 43 = DeletedRecordLocator, 44 = UpgradeClassOfService, 45 = DowngradeClassOfService, 46 = StandbyPriorityChange, 47 = AssignedTicketNumber, 48 = DeletedTicketNumber, 49 = ConfirmSegmentStatusCodeChange, 50 = CodeshareFlightChanged, 51 = PdsCancel, 52 = PdsPending, 53 = PdsConfirm, 54 = PdsFinalized, 55 = PdsDeclined, 56 = PdsException, 57 = PdsCancelRefused, 58 = PdsCancelUnsuccessful, 59 = Apps, 60 = InhibitedOverride, 61 = PrintedBagTag, 62 = SelfPrintedBagTag, 63 = PrintedBoardingPass, 64 = AddCustomerId, 65 = DeleteCustomerId, 66 = HoldCreated, 67 = HoldRemoved, 68 = HoldChanged, 69 = OverrideCoupon, 70 = PdsSynchronized, 71 = PdsItemremoved, 72 = Reprice, 73 = ChannelOverride, 74 = EmdCreated, 75 = EmdRemoved, 76 = EmdChanged, 77 = ServiceBundle, 78 = PublishedFareOverride, 79 = FareClassRealignment, 80 = AddedDocument, 81 = ChangedDocument || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The booking history.

400 Response

401 Response

404 Response

Examples

Response 200

Retrieve Booking History Response:

```
json   
{  
    "data": {  
        "histories": [  
            {  
                "historyCategory": 30,  
                "details": "AD P 1234",  
                "event": 2,  
                "pointOfSale": {  
                    "isoCountryCode": null,  
                    "sourceSystemCode": null,  
                    "agentCode": "CCUserM",  
                    "domainCode": "DEF",  
                    "locationCode": "HDQ",  
                    "organizationCode": "SYSTEM"  
                },  
                "sourcePointOfSale": {  
                    "isoCountryCode": null,  
                    "sourceSystemCode": null,  
                    "agentCode": "CCUserM",  
                    "domainCode": "DEF",  
                    "locationCode": "HDQ",  
                    "organizationCode": "SYSTEM"  
                },  
                "receivedBy": "Dailey, Paltry",  
                "receivedByReference": "801-555-9965",  
                "createdDate": "2025-08-07T04:47:16.3456254+00:00"  
            }  
        ],  
        "pageKey": null  
    }  
}
```

Description: Example response for successfully retrieving the booking history.

With Invalid Event Type:

```

json

{
  "messages": [
    {
      "rawValue": "The value '83' is invalid.",
      "code": "core:Validation:MalformedRequest",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "Information",
      "value": "core:Validation:MalformedRequest",
      "status": 0,
      "details": null
    }
  ],
  "data": {
    "histories": [
      {
        "historyCategory": 30,
        "details": "AD P 1234",
        "event": 2,
        "pointOfSale": {
          "isoCountryCode": null,
          "sourceSystemCode": null,
          "agentCode": "CCUserM",
          "domainCode": "DEF",
          "locationCode": "HDQ",
          "organizationCode": "SYSTEM"
        },
        "sourcePointOfSale": {
          "isoCountryCode": null,
          "sourceSystemCode": null,
          "agentCode": "CCUserM",
          "domainCode": "DEF",
          "locationCode": "HDQ",
          "organizationCode": "SYSTEM"
        },
        "receivedBy": "Dailey, Paltry",
        "receivedByReference": "801-555-9965",
        "createdDate": "2025-08-07T04:47:16.3456349+00:00"
      }
    ],
    "pageKey": null
  }
}

```

Description: Example response if event enum is invalid.

Response 400

With No Committed Booking:

```

json

{
  "errors": [
    {
      "id": "eb0e3db1-a704-6c9b-d89b-bfd4d662af74",
      "code": "nsk:Booking:CommittedStateRequired",
      "message": "nsk:Booking:CommittedStateRequired",
      "type": "Validation",
      "details": null,
      "rawMessage": "Invalid state. The booking should be committed before this operation.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}

```

Description: Example error response if there is no committed booking.

GET /api/nsk/v1/booking/history/bagTagPrint

Summary: Gets the current bag tag print history for the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
isBagTagSelfPrinted	query	boolean	No	Whether the bag tag is self printed.

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/history/bagTagPrint

Summary: Creates a bag tag print history for the booking in state.

Description

Also updates bag's status from Added to AddedPrinted if Self-Tagging Baggage is enabled. In order to see the status of a bag, check bags on a passenger segment.

GraphQL endpoints: bookingBagTagPrintHistoryAdd, bookingBagTagPrintHistoryAddBatch

Request Details

Request Body

The bag tag print history request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/booking/history/flightMove

Summary: Gets the current flight move history in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
LastPageKey	query	string	No	The key of the last booking history item returned.

plaintext ✖

|

| PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The flight move history response.

404 Response

GET /api/nsk/v1/booking/history/holdDateChange

Summary: Gets the current hold date change history for the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Event	query	string	No	The event that triggered the hold date change history entry.

Enumeration values: 66 = Created, 67 = Removed, 68 = Changed | | LastPageKey | query | string | No | The key of the last booking history item returned. | | PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The hold date change history response.

404 Response

GET /api/nsk/v1/booking/history/itinerarySent

Summary: Gets the current itinerary sent history for the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
LastPageKey	query	string	No	The key of the last booking history item returned.

plaintext ✖

|

| PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The itinerary sent history response.

404 Response

GET /api/nsk/v1/booking/history/message

Summary: Gets the current message history in state.

Description

Request Details

Responses

200 Response

The list of message history.

404 Response

GET /api/nsk/v1/booking/history/notification

Summary: Gets the current notification history in state.

Description

Request Details

Responses

200 Response

The list of notification history.

404 Response

GET /api/nsk/v1/booking/history/seatAssignment

Summary: Gets the current seat assignment history in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Event	query	string	No	The event that triggered the seat assignment history entry.

Enumeration values: 4 = AssignedSeat, 5 = RemoveSeat || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The seat assignment history response.

404 Response

GET /api/nsk/v1/booking/history/segmentChange

Summary: Gets the current segment change history in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Event	query	string	No	The event that triggered the seat assignment history entry.

Enumeration values: 6 = AddedFlight, 7 = DeletedFlight || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

The segment change history response.

404 Response

Booking/Journeys Endpoints

GET /api/nsk/v1/booking/journeys

Summary: Gets the collection of journeys on the booking in state.

Description

Request Details

Responses

200 Response

The collection of journeys.

404 Response

DELETE /api/nsk/v1/booking/journeys

Summary: Cancels all the journeys on the current booking in state.

Description

This endpoint also cancels all booking components/add-ons. To only cancel journeys, use DELETE /api/nsk/v1/booking/journeys/{journeyKey}.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

No content.

GET /api/nsk/v1/booking/journeys/{journeyKey}

Summary: Gets a single journey by the key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	No description

Responses

200 Response

Ok.

404 Response

Not found.

DELETE /api/nsk/v1/booking/journeys/{journeyKey}

Summary: Cancels the specific journey from the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key to be canceled.

Request Body

The cancel journey request options.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Metadata response message.

DELETE /api/nsk/v1/booking/journeys/{journeyKey}/bundles

Summary: Removes the add-on service bundles of the requested passengers in the given journey.

Description

This endpoint does not support removal of included bundles.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.

Request Body

The passengers to remove bundles from.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

404 Response

200 Response

POST /api/nsk/v2/booking/journeys/{journeyKey}/bundles

Summary: Sells a service bundle for one or more passengers on a journey.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.

Request Body

The bundle sell request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Examples

Response 201

With Already Booked SSR or Bundle:

```
json
{
  "messages": [
    {
      "rawValue": "[HMC] cannot be sold because passenger 0 already booked [THCS] on NV3075 YTZ-LAX.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "[HMC] cannot be sold because passenger 0 already booked [THCS] on NV3075 YTZ-LAX."
      }
    },
    "data": null
  ]
}
```

Description: Example OSI Message if passenger already booked an SSR or bundle

With Cannot Find Interchangeable Bundle SSR:

```
json
{
  "messages": [
    {
      "rawValue": "Cannot find interchangeable bundle SSR for [HMC] in state.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "Cannot find interchangeable bundle SSR for [HMC] in state."
      }
    },
    "data": null
  ]
}
```

Description: Example OSI Message if cannot find interchangeable bundle SSR in state

With SSR Not Found in Service Bundle:

```
json
{
  "messages": [
    {
      "rawValue": "SSR [HMC] not found in service bundle.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "SSR [HMC] not found in service bundle."
      }
    },
    "data": null
  }
}
```

Description: Example OSI Message if an SSR is not found in service bundle

With Duplicate SSR or Bundle:

```
json
{
  "messages": [
    {
      "rawValue": "Duplicate SSR [THCS] in SAN-YTZ is not allowed for service bundle.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "Duplicate SSR [THCS] in SAN-YTZ is not allowed for service bundle."
      }
    },
    "data": null
  }
}
```

Description: Example OSI Message if duplicate SSR or Bundle

DELETE /api/nsk/v1/booking/journeys/{journeyKey}/fees/penalty

Summary: Waives the penalty fee of the specified journey.

Description

Waiving the penalty fee is configurable. See the "Fee and Tax Settings", "Single Penalty Fee" option. If this is checked and configured this endpoint will waive the penalty fees from all journeys in the booking in state.

```
plaintext
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.

Responses

200 Response

GET

/api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/address/requirements

Summary: Gets the passenger or associated infant's destination address requirements for a specific journey for the booking in-state.

Description

This endpoint will determine if the passenger or the associated infant requires a destination address for the given journey. This does not determine if the passenger already has a qualifying address, only if the destination address is required.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key to check requirements for.
passengerKey	path	string	Yes	The passenger key to check requirements for.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage

Summary: Gets all baggage for a passenger on a specific journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key for the journey to retrieve from.
passengerKey	path	string	Yes	The passenger key for the passenger to retrieve from.

Responses

200 Response

OK.

404 Response

Baggage was not found.

POST /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage

Summary: Adds baggage to the booking.

Description

This call must be followed by a commit to see the bags added in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique identifier for the journey to add to.
passengerKey	path	string	Yes	The specific passenger key.

Request Body

A request to add bags to the given passenger.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET

/api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey}

Summary: Gets a single bag from a passenger on a journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key for the journey to retrieve from.
passengerKey	path	string	Yes	The passenger key for the passenger to retrieve from.
baggageKey	path	string	Yes	The bag key for the bag to retrieve.

Responses

200 Response

OK.

404 Response

Bag was not found.

DELETE

/api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey}

Summary: Removes baggage from the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique identifier for the journey to remove from.
passengerKey	path	string	Yes	The specific passenger key.
baggageKey	path	string	Yes	Unique identifier for the bag to remove.

Responses

200 Response

Ok.

PUT

/api/nsk/v2/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey}

Summary: Updates the baggage for a passenger on the booking in state.

Description

This is to be called after a stateful adding of baggage to a booking and a commit must follow the completion of this in order to see the bag checked in. This call does NOT sync with baggage allowances and usages.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
passengerKey	path	string	Yes	The passenger key.
baggageKey	path	string	Yes	The baggage key.

Request Body

The baggage update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Response 400

With Invalid Baggage Key:

json



```
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "baggageKey": "MDAyODAzMDg0MQ--"
      },
      "rawMessage": "The identifier 'BaggageKey' with value 'MDAyODAzMDg0MQ--' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example Error if baggage key is invalid.

Update Failed Exception:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:UpdateFailed",
      "message": "nsk:Exceptions:UpdateFailed",
      "type": "Validation",
      "details": null,
      "rawMessage": "Attempted update of object PassengerBagUpdateRequest has failed. Please verify this object exists.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error if the update failed.

PUT

/api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/{baggageKey}/checkIn

Summary: Checks in the baggage for a passenger on the booking in state.

Description

This is to be called after a stateful adding of baggage to a booking and a commit must follow the completion of this in order to see the bag checked in. Also, this call properly syncs with baggage allowances and usages once completed.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
passengerKey	path	string	Yes	The passenger key.
baggageKey	path	string	Yes	The baggage key.

Request Body

The baggage check-in request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Response 400

With Invalid Baggage Key:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "baggageKey": "MDAYODAzMDg0MQ--"
      },
      "rawMessage": "The identifier 'BaggageKey' with value 'MDAYODAzMDg0MQ--' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if baggage key is invalid.

Baggage Checkin Failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Baggage:BaggageCheckinFailed",
      "message": "nsk:Baggage:BaggageCheckinFailed",
      "type": "Error",
      "details": null,
      "rawMessage": "The baggage check-in for the booking in state failed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: If baggage check-in failed.

PUT /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/group

Summary: Moves passenger to a different baggage group.

Description

Passengers in the same baggage group may share baggage allowances, enabling baggage allowance pooling.

`PassengerBagGroupRequest.BaggageGroupNumber` only accepts bag group numbers from existing passenger bag group numbers. Passenger bag group number is exposed under passenger segment. If set to null, the passenger will be moved to a new bag group.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

GraphQL endpoints: `passengerBagGroupSet`, `passengerBagGroupSetBatch`

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
passengerKey	path	string	Yes	The passenger key.

Request Body

The passenger bag group request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

POST /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/baggage/manual

Summary: Adds manually tagged baggage to the booking.

Description

This call must be followed by a commit to see the bags added in state.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique identifier for the journey to add to.
passengerKey	path	string	Yes	The specific passenger key.

Request Body

A request to add manually tagged bags to the given passenger.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

DELETE /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/fees/amount

Summary: Waives fees for a specific journey and passenger, based upon the given fee type. This is an agent-only endpoint.

Description

NOTE: This endpoint takes a singular property for the request body and as such the request will need to be formatted differently than what it typically looks like. For more information, please see <https://navitaire-developer-digital-api.azurewebsites.net/articles/Miscellaneous/UniqueInputSingleRequestBodyREST>. and the Swagger request examples.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
journeyKey	path	string	Yes	The journey key.

Request Body

The waive fee type.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Examples

Request

Waiving a fee of type Spoilage Fee JSON request body, 1:

```
json
```

```
2
```



Description: Example request for when wanting to have a waive fee type of SpoilageFee .

Waiving a fee of type Spoilage Fee JSON request body, 2:

```
json
```

```
"SpoilageFee"
```



Description: Example request for when wanting to have a waive fee type of SpoilageFee .

POST /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/infant

Summary: Creates an infant on a passenger in the specified journey.

Description

If the booking in state contains a passive segment, an infant will not be created on the passive segment for the passenger. If one of the segments in the journey already has an infant, the infant details from the request will be ignored and the infant details from the existing infant will be retained.

```
plaintext
```



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique journey key.
passengerKey	path	string	Yes	The unique passenger key.

Request Body

The new infant.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

DELETE /api/nsk/v1/booking/journeys/{journeyKey}/passengers/{passengerKey}/infant

Summary: Deletes an infant from a passenger in the specified journey.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique identifier of the journey to remove infants from.
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

PATCH /api/nsk/v1/booking/journeys/{journeyKey}/segments/{segmentKey}/status

Summary: Modifies the status of the specified segment in the journey.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey that contains the specified segment.
segmentKey	path	string	Yes	The segment to be modified.
newStatus	query	string	No	The new segment status to be set.
Enumeration values: 0 = UnknownStatus, 1 = NoActionTaken, 2 = UnableToConfirmSegment, 3 = HeldConfirmed, 4 = HeldCancelled, 5 = ConfirmedTimeChange				

Responses

200 Response

GET /api/nsk/v1/booking/journeys/{journeyKey}/travelDocument/requirements

Summary: Gets the travel document requirements for a specific journey for the booking in-state.

Description

This endpoint will return a response containing a collection of travel document type codes that could fulfill a travel document requirement. If the "AtLeastOneDocumentRequired" value is true, then one or more documents in the list are required. However, it does not mean that every document in the list is required. If "AtLeastOneDocumentRequired" is false, then the list of document type codes represents a list of acceptable travel documents, but nothing is required. This endpoint does not determine if any passengers already have required travel documents.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key to get required travel documents for.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/journeys/passengers/{passengerKey}/address/requirements

Summary: Gets the passenger or associated infant's destination address requirements for all journeys for the booking in-state.

Description

This endpoint will determine if the passenger or the associated infant requires a destination address. If a destination address is required for the passenger or associated infant for any journey, then this will return true. This does not determine if the passenger already has a qualifying address, only if the destination address is required.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key to check requirements for.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/journeys/travelDocument/requirements

Summary: Gets the travel document requirements for all journeys for the booking in-state.

Description

This endpoint will return a response containing a collection of travel document type codes that could fulfill a travel document requirement. If the "AtLeastOneDocumentRequired" value is true, then one or more documents in the list are required. However, it does not mean that every document in the list is required. If "AtLeastOneDocumentRequired" is false, then the list of document type codes represents a list of acceptable travel documents, but nothing is required. This endpoint does not determine if any passengers already have required travel documents.

Request Details

Responses

200 Response

404 Response

Booking/Passengers Endpoints

GET /api/nsk/v1/booking/passengers

Summary: Gets the collection of passengers on the booking.

Description

Request Details

Responses

200 Response

The collection of passengers.

404 Response

GET /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications

Summary: Gets passenger travel notifications for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications

Summary: Creates a passenger travel notification for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.

Request Body

The passenger travel notification create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe}

Summary: Gets a passenger travel notification for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

400 Response

404 Response

PUT

[`/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe`](#)

Summary: Updates a passenger travel notification for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

PATCH

[`/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe`](#)

Summary: Patches a passenger travel notification for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The delta mapper travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

DELETE

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Deletes a passenger travel notification for a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

400 Response

GET

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Gets notification events for a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Adds a notification event to a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification event create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Get a notification event for a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

404 Response

DELETE

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Deletes a notification event from a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The notification event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

GET

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Gets notification timed events for a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe

Summary: Adds a notification timed event to a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification timed even create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe}

Summary: Get a notification timed event for a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

404 Response

PUT

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe}

Summary: Updates a notification timed event for a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Request Body

The updated notification timed event request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE

/api/nsk/v1/booking/passengers/{passengerAlternateKey}/travelNotifications/{travelNotificationKe}

Summary: Deletes a notification timed event from a specific passenger travel notification on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerAlternateKey	path	string	Yes	The passenger alternate key. This key will be null until the booking is committed.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}

Summary: Gets a specific passenger on the booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

The passenger.

404 Response

PUT /api/nsk/v3/booking/passengers/{passengerKey}

Summary: Updates a specific passenger on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

GraphQL endpoints: passengerSetv3, passengersSetBatch

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
waiveNameChangeFees	query	boolean	No	Waive the name change fees.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel documents' gender. This will update the gender of all the travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.

```
plaintext
|
|
```

Request Body

The modified passenger request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

400 Response

409 Response

Examples

Response 200

Passenger Updated:

```
json
{
  "data": null
}
```

Description: The passenger has been updated.

With mismatching passenger and travel document gender:

```
json
{
  "messages": [
    {
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",
      "code": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "FailedTravelDocValidation",
      "value": null,
      "status": 0,
      "details": null
    }
  ],
  "data": null
}
```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

PATCH /api/nsk/v3/booking/passengers/{passengerKey}

Summary: Patches a specific passenger on the booking.

Description

Please note, the API will perform validation for the same properties that are on both the PATCH request and the existing passenger.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

GraphQL endpoints: `passengerModifyv3`, `passengersModifyBatch`

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
waiveNameChangeFees	query	boolean	No	Waive the name change fee.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel
documents' gender. This will update the gender of all the travel documents in the				
passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.				

plaintext



Request Body

The patched passenger request.

Content Type: `application/json`

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

400 Response

409 Response

Examples

Response 200

Passenger Updated:

```
json
{
  "data": null
}
```

Description: The passenger has been updated.

With mismatching passenger and travel document gender:

```
json
{
  "messages": [
    {
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",
      "code": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "FailedTravelDocValidation",
      "value": null,
      "status": 0,
      "details": null
    }
  ],
  "data": null
}
```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

GET /api/nsk/v1/booking/passengers/{passengerKey}/addresses

Summary: Gets the collection of addresses from a specific passenger.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

The collection of passenger addresses.

404 Response

POST /api/nsk/v2/booking/passengers/{passengerKey}/addresses

Summary: Creates a new address for a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Request Body

The new address request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/booking/passengers/{passengerKey}/addresses/{addressKey}

Summary: Gets a specific address from a specific passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
addressKey	path	string	Yes	The unique address key.

Responses

200 Response

The address.

404 Response

PUT /api/nsk/v3/booking/passengers/{passengerKey}/addresses/{addressKey}

Summary: Updates a specific address on a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
addressKey	path	string	Yes	The unique address key.

Request Body

The modified address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v3/booking/passengers/{passengerKey}/addresses/{addressKey}

Summary: Patches a specific address on a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
addressKey	path	string	Yes	The unique address key.

Request Body

The patched address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v2/booking/passengers/{passengerKey}/addresses/{addressKey}

Summary: Deletes a specific address on a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
addressKey	path	string	Yes	The unique address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/booking/passengers/{passengerKey}/baggage

Summary: Gets all the baggage for a specified passenger.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique identifier for the passenger to get baggage for.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}

Summary: Retrieve a single bag from a specific passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key for the passenger to retrieve from.
baggageKey	path	string	Yes	The bag key for the bag to retrieve.

Responses

200 Response

OK.

404 Response

Bag was not found.

GET /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events

Summary: Retrieves events for a specific passenger baggage.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Passenger key.
baggageKey	path	string	Yes	Baggage key.
StartDate	query	string (date-time)	No	Start date. If no start date is provided, the start date defaults to the current UTC-based date.

plaintext 

|

| EndDate | query | string (date-time) | No | End date. If no end date is provided, the end date defaults to the current UTC-based date. |

Responses

200 Response

500 Response

POST /api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events

Summary: Creates event information for a specific passenger baggage. A successful call immediately creates the event and does not require the booking to be committed.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Passenger key.
baggageKey	path	string	Yes	Baggage key.

Request Body

Baggage event create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

500 Response

GET

/api/nsk/v1/booking/passengers/{passengerKey}/baggage/{baggageKey}/events/{baggageEventKe

Summary: Retrieves information for a specific passenger baggage event.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Passenger key.
baggageKey	path	string	Yes	Baggage key.
baggageEventKey	path	string	Yes	Baggage event key.
StartDate	query	string (date-time)	No	Start date. If no start date is provided, the start date defaults to the current UTC-based date.

plaintext



|

| EndDate | query | string (date-time) | No | End date. If no end date is provided, the end date defaults to the current UTC-based date. |

Responses

200 Response

500 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/breakdown

Summary: Gets a specific passenger price breakdown by passenger key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The specific passenger key.

Responses

200 Response

The passenger price breakdown.

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/documents

Summary: Gets a specific passenger's collection of travel documents.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

The collection of passenger travel documents.

404 Response

POST /api/nsk/v2/booking/passengers/{passengerKey}/documents

Summary: Create a new travel document for a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel
documents' gender. This will update the gender in the passengers[passengerKey] as well as the gender of the other				
travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.				

plaintext

|

Request Body

The new travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

409 Response

Examples

Response 201

Passenger Travel Document Created:

```
json   
{  
  "data": {  
    "passengerTravelDocumentKey": "UCFVUyEg"  
  }  
}
```

Description: The passenger travel document has been created.

With mismatching passenger and travel document gender:

```
json   
{  
  "messages": [  
    {  
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",  
      "code": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",  
      "type": "FailedTravelDocValidation",  
      "value": null,  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": {  
    "passengerTravelDocumentKey": "UCFVUyEg"  
  }  
}
```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

GET /api/nsk/v1/booking/passengers/{passengerKey}/documents/{travelDocumentKey}

Summary: Gets a specific passenger's specific travel document.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.

Responses

200 Response

The travel document.

404 Response

PUT /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey}

Summary: Updates a specific passenger's travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel documents' gender. This will update the gender in the passengers[passengerKey] as well as the gender of the other travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.

plaintext



|

Request Body

The modified travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The updated travel document.

500 Response

409 Response

Examples

Response 200

Passenger Travel Document Updated:

json



```
{  
  "data": null  
}
```

Description: The passenger travel document has been updated.

With mismatching passenger and travel document gender:

```

json

{
  "messages": [
    {
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",
      "code": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "FailedTravelDocValidation",
      "value": null,
      "status": 0,
      "details": null
    }
  ],
  "data": null
}

```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

PATCH /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey}

Summary: Patches a specific passenger's travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel
documents' gender. This will update the gender in the passengers[passengerKey] as well as the gender of the other travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.				

```

plaintext
|
```

Request Body

The patched travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

409 Response

Examples

Response 200

Passenger Travel Document Updated:

```
json

{
  "data": null
}
```

Description: The passenger travel document has been updated.

With mismatching passenger and travel document gender:

```
json

{
  "messages": [
    {
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",
      "code": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "FailedTravelDocValidation",
      "value": null,
      "status": 0,
      "details": null
    }
  ],
  "data": null
}
```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

DELETE /api/nsk/v2/booking/passengers/{passengerKey}/documents/{travelDocumentKey}

Summary: Deletes a specific passenger's specific travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.

Responses

200 Response

Ok.

GET /api/nsk/v1/booking/passengers/{passengerKey}/documents/check

Summary: Gets the document check status for a specific passenger if there are any.

Description

Requires a committed booking in state. Requires document check to be enabled in government security configuration.

Performs document check and returns the results as `documentCheckStatus`. More detailed information is intentionally not provided for security reasons. The designers of the client application will need to determine what to do with the different responses, in most cases the passenger will

need to be assisted by an airport agent. For more specific requirements check the following endpoint `GET/api/nsk/v1/booking/checkin/journey/{journeyKey}/requirements`. For configuration options please contact your CSDM. For full documentation see <https://developer.navitaire.com/documentation/dotrez-api/how-to/adc>.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The key of the passenger to perform the document check on.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/emdCoupons

Summary: Gets the collection of passenger EMD coupons for a specific passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/emdCoupons/{passengerEmdCouponKey}

Summary: Gets a specific passenger EMD coupon for a specific passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
passengerEmdCouponKey	path	string	Yes	The unique passenger EMD coupon key.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/fees

Summary: Gets all fees for a specific passenger, based upon the given passenger key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 200

Retrieve passenger fees response:

```
json 
```

```
{  
    "data": [  
        {  
            "type": 7,  
            "ssrCode": null,  
            "ssrNumber": 0,  
            "paymentNumber": 0,  
            "isConfirmed": false,  
            "isConfirming": true,  
            "isConfirmingExternal": false,  
            "code": "NFSF",  
            "detail": null,  
            "passengerFeeKey": "MCEwITc-",  
            "override": false,  
            "flightReference": null,  
            "note": "b",  
            "createdDate": "2025-03-26T05:24:42.7610199+00:00",  
            "isProtected": false,  
            "serviceCharges": [  
                {  
                    "amount": 0.0,  
                    "code": "NFSF",  
                    "detail": "for free",  
                    "type": 6,  
                    "collectType": 0,  
                    "currencyCode": "USD",  
                    "foreignCurrencyCode": "USD",  
                    "foreignAmount": 0.0,  
                    "ticketCode": "NSF"  
                }  
            ]  
        },  
        {  
            "type": 5,  
            "ssrCode": null,  
            "ssrNumber": 0,  
            "paymentNumber": 0,  
            "isConfirmed": false,  
            "isConfirming": true,  
            "isConfirmingExternal": false,  
            "code": "CHF1",  
            "detail": "26Mar NV2033 SLCDEN",  
            "passengerFeeKey": "MCExITU-",  
            "override": false,  
            "flightReference": "20250326 NV2033 SLCDEN",  
            "note": "a",  
            "createdDate": "2025-03-26T05:27:52.1473296+00:00",  
            "isProtected": true,  
            "serviceCharges": [  
                {  
                    "amount": 50.0,  
                    "code": "CHF1",  
                    "detail": "Change Fee",  
                    "type": 6,  
                    "collectType": 0,  
                    "currencyCode": "USD",  
                    "foreignCurrencyCode": "USD",  
                    "foreignAmount": 50.0,  
                    "ticketCode": "CHF"  
                }  
            ]  
        }  
    ]  
}
```

Description: Example response for successfully retrieving the passenger fees.

No passenger fees response:

```
json

{
    "data": []
}
```

Description: Example response when there are no fees associated with the passenger.

Response 400

No booking in state:

```
json

{
    "errors": [
        {
            "id": "4d8dbf9a-ea54-6c63-e755-da5c0328543d",
            "code": "nsk:Booking:NoBookingInState",
            "message": "nsk:Booking:NoBookingInState",
            "type": "Validation",
            "details": null,
            "rawMessage": "No booking was found in session, cannot perform operation.",
            "exception": null,
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
        }
    ],
    "data": null
}
```

Description: Example error when there is no booking in state.

Invalid passenger key:

```
json

{
    "errors": [
        {
            "id": "1c2e93af-10dd-3da9-56ae-d1e50e731906",
            "code": "nsk:Exceptions:InvalidKey",
            "message": "The identifier 'passengerKey' with value 'MCF' is invalid.",
            "type": "Validation",
            "details": null,
            "rawMessage": "The identifier 'passengerKey' with value 'MCF' is invalid.",
            "exception": null,
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
        }
    ],
    "data": null
}
```

Description: Example error when the passenger key is invalid or does not exist in the booking in state.

GET /api/nsk/v1/booking/passengers/{passengerKey}/fees/{passengerFeeKey}

Summary: Gets a specific fee for a specific passenger, based upon the given keys.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
passengerFeeKey	path	string	Yes	The passenger fee key.

Responses

200 Response

404 Response

DELETE /api/nsk/v1/booking/passengers/{passengerKey}/fees/nameChange

Summary: Waives the name change fee of the specified passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.

Responses

200 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/fees/quote

Summary: Gets the override fee quote of a new fee.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
FeeCode	query	string	No	The fee code used to identify the fee being overridden.

Currently, only manually-added fees are supported for quoting new fee overrides. These fee types are:

- Service Fee
- Non-Flight Service Fee
- Penalty Fee
- Spoilage Fee |

| PricingMode | query | string | No | Indicates if fees are to be paid in points or with currency

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly || OriginStation | query | string | No | Originating station code of the booking where the fee will be applied. || Amount | query | number (decimal) | No | The override amount. || CollectedCurrencyCode | query | string | No | The collected currency code. |

Responses

200 Response

400 Response

404 Response

Examples

Response 200

Retrieve Override Fee Quote Response:

```
json
{
  "data": {
    "feeOverrideCharges": [
      {
        "amount": 100.0,
        "code": "NFSF",
        "detail": "Non Flight",
        "type": 6,
        "collectType": 0,
        "currencyCode": "USD",
        "foreignCurrencyCode": "USD",
        "foreignAmount": 100.0,
        "ticketCode": "NSF"
      }
    ]
  }
}
```

Description: Example response for successfully retrieving the override fee quote response.

Response 400

With No Booking in State:

```
json
{
  "errors": [
    {
      "id": "505b55d3-cac8-93a9-4d25-131256cdd551",
      "code": "nsk:Booking:NoBookingInState",
      "message": "nsk:Booking:NoBookingInState",
      "type": "Validation",
      "details": null,
      "rawMessage": "No booking was found in session, cannot perform operation.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if there is no booking in state.

With Invalid Passenger Key:

```
json
{
  "errors": [
    {
      "id": "2987bed7-5690-5050-55e9-819b8551b344",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'PassengerKey' with value 'MCFBASDFRFQ-' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if passenger key is invalid.

With Invalid Currency Code:

```
json
{
  "errors": [
    {
      "id": "10daf3f4-82c2-cf99-21ba-169fadae0b10",
      "code": "nsk:Validation:UnknownSellCurrencyCode",
      "message": "nsk:Validation:UnknownSellCurrencyCode",
      "type": "Validation",
      "details": {
        "currencyCode": "SDK"
      },
      "rawMessage": "The currency code SDK does not exist or is not active.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the currency code is invalid.

With Invalid Fee Key:

```
json
{
  "errors": [
    {
      "id": "ea9e908c-8979-85d8-c9ed-886014d6fc46",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'FeeKey' with value 'MT21XF' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if fee key is invalid.

With Invalid Fee Code:

```
json
{
  "errors": [
    {
      "id": "bf7c1a48-a4bf-ddf7-2413-93959ba6a3b3",
      "code": "nsk-server:FeeCalculation",
      "message": "nsk-server:FeeCalculation",
      "type": "Validation",
      "details": null,
      "rawMessage": "Cannot quote overrides of type NameChangeFee for new fees.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee code is not allowed to be quoted.

With Null Fee Code:

```
json
{
  "errors": [
    {
      "id": "6b34775e-d264-f3a6-4034-62daf6699617",
      "code": "GetNewOverrideFeeQuoteRequestBase.FeeCode:RequiredAttribute",
      "message": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "GetNewOverrideFeeQuoteRequestBase",
        "member": "FeeCode"
      },
      "rawMessage": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee code is null

With Null Fee Amount:

```
json
{
  "errors": [
    {
      "id": "6b34775e-d264-f3a6-4034-62daf6699617",
      "code": "GetNewOverrideFeeQuoteRequestBase.FeeCode:RequiredAttribute",
      "message": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "GetNewOverrideFeeQuoteRequestBase",
        "member": "Amount"
      },
      "rawMessage": "The GetNewOverrideFeeQuoteRequestBase field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee amount is null

With Allow Negative Fees Role Setting Off:

```
json
{
  "errors": [
    {
      "id": "a49a0074-9b6d-9e30-83c3-1c21a56d748c",
      "code": "nsk:Validation:RoleSettingsProhibit",
      "message": "nsk:Validation:RoleSettingsProhibit",
      "type": "Validation",
      "details": null,
      "rawMessage": "Role settings prohibit a negative fee.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the fee amount is negative or the fee type is not allowed to override, and the role setting is turned off.

With Inactive Fees:

```
json

{
  "errors": [
    {
      "id": "bee189bc-edfc-263b-edc1-33cbf8b709cf",
      "code": "nsk:Fee:NotFound",
      "message": "nsk:Fee:NotFound",
      "type": "Validation",
      "details": null,
      "rawMessage": "The fee was not found.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the agent attempted to sell inactive fees.

With Unauthorized Overriding of Service Fees:

```
json

{
  "errors": [
    {
      "id": "7911302f-46af-3dc8-5714-b2823fec853d",
      "code": "nsk-server:AuthorizationSSRNotAllowed",
      "message": "Agent is not authorized to override fees.",
      "type": "Validation",
      "details": {
        "message": "Agent is not authorized to override fees."
      },
      "rawMessage": "Agent is not authorized to override fees.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the agent is not authorized to override a service fee.

With Unauthorized Selling of Service Fees:

```
json

{
  "errors": [
    {
      "id": "cfdacd38-5e2c-ab09-9fce-86722e945f0b",
      "code": "nsk-server:AuthorizationServiceFeeNotAllowed",
      "message": "Agent is not authorized to sell service fee FLO.",
      "type": "Validation",
      "details": {
        "message": "Agent is not authorized to sell service fee FLO."
      },
      "rawMessage": "Agent is not authorized to sell service fee FLO.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example Error if the agent is not authorized to sell a service fee.

GET /api/nsk/v1/booking/passengers/{passengerKey}/infant

Summary: Gets a specific passenger's infant.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

The passenger infant.

404 Response

POST /api/nsk/v3/booking/passengers/{passengerKey}/infant

Summary: Creates an infant on a specific passenger.

Description

If the booking in state contains a passive segment, an infant will not be created on the passive segment for the passenger.

plaintext	
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.	

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Request Body

The new infant.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

PUT /api/nsk/v3/booking/passengers/{passengerKey}/infant

Summary: Updates an infant on a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key
syncInfantAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the infant and all its travel documents' gender. This will update the gender of
all the travel documents in the passengers[passengerKey].infant.travelDocuments list in the booking model. The value defaults to true if not specified.				

```
plaintext 
```

Request Body

The modified infant.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

500 Response

A server error occurred.

409 Response

Concurrent changes were being made in the same session.

PATCH /api/nsk/v2/booking/passengers/{passengerKey}/infant

Summary: Patches an infant on a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
syncInfantAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the infant and all its travel documents' gender. This will update the gender of
all the travel documents in the passengers[passengerKey].infant.travelDocuments list in the booking model. The value defaults to true if not specified.				

```
plaintext 
```

Request Body

The patched infant.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

500 Response

A server error occurred.

409 Response

Concurrent changes were being made in the same session.

DELETE /api/nsk/v2/booking/passengers/{passengerKey}/infant

Summary: Deletes an infant from a specific passenger.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

OK. Request completed successfully.

409 Response

Concurrent changes were being made in the same session.

GET /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents

Summary: Gets a specific passenger infant's collection of travel documents.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

The collection of passenger infant travel documents.

404 Response

POST /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents

Summary: Create a new travel document for a specific passenger infant.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
syncInfantAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the infant and all its travel documents' gender. This will update the gender in
the passengers[passengerKey].infant as well as the gender of the other travel documents in the				
passengers[passengerKey].infant.travelDocuments list in the booking model. The value defaults to true if not specified.				

plaintext Copy

```
|
```

Request Body

The new travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

409 Response

Concurrent changes were being made in the same session.

GET /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey}

Summary: Gets a specific passenger infant's specific travel document.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.

Responses

200 Response

The travel document.

404 Response

PUT /api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey}

Summary: Updates a specific passenger infant travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.
syncInfantAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the infant and all its travel documents' gender. This will update the gender in
the passengers[passengerKey].infant as well as the gender of the other travel documents in the				
passengers[passengerKey].infant.travelDocuments list in the booking model. The value defaults to true if not specified.				

plaintext Copy

```
|
```

Request Body

The modified travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

500 Response

A server error occurred.

409 Response

Concurrent changes were being made in the same session.

PATCH

/api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey}

Summary: Patches a specific passenger infant travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.
syncInfantAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the infant and all its travel documents' gender. This will update the gender in the passengers[passengerKey].infant as well as the gender of the other travel documents in the passengers[passengerKey].infant.travelDocuments list in the booking model. The value defaults to true if not specified.

```
plaintext 
```

Request Body

The patched travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

500 Response

A server error occurred.

409 Response

Concurrent changes were being made in the same session.

DELETE

/api/nsk/v2/booking/passengers/{passengerKey}/infant/documents/{travelDocumentKey}

Summary: Deletes a specific passenger infant specific travel document.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
travelDocumentKey	path	string	Yes	The unique travel document key.

Responses

200 Response

OK. Request completed successfully.

409 Response

Concurrent changes were being made in the same session.

GET /api/nsk/v1/booking/passengers/{passengerKey}/infant/documents/check

Summary: Gets the document check status for a specific passenger's infant if there are any.

Description

Requires a committed booking in state. Requires document check to be enabled in government security configuration.

Performs document check and returns the results as documentCheckStatus. More detailed information is intentionally not provided for security reasons. The designers of the client application will need to determine what to do with the different responses, in most cases the passenger will need to be assisted by an airport agent. For more specific requirements check the following endpoint

GET/api/nsk/v1/booking/checkin/journey/{journeyKey}/requirements. For configuration options please contact your CSDM. For full documentation see <https://developer.navitaire.com/documentation/dotrez-api/how-to/adc>.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The key of the passenger to perform the infant document check on.

Responses

200 Response

404 Response

PATCH /api/nsk/v1/booking/passengers/{passengerKey}/passengerTypeCode

Summary: Patches a specific passenger to the given passenger type.

Description

NOTE: The date of birth doesn't need to be updated if the passenger's current date of birth still falls within the new passenger type's age restriction. This could also cause a repricing on the booking. In addition, changing the passenger type will change the passenger key as well.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The unique passenger key.
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel
documents' gender. This will update the gender of all the travel documents in the				
passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.				

plaintext



|

Request Body

The passenger type request containing the new passenger type.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

500 Response

A server error occurred.

409 Response

Examples

Response 200

Passenger Type Code Modified:

```
json   
{  
  "data": {  
    "oldPassengerKey": "MCFBRFQ-",  
    "newPassengerKey": "MCFTT1I-"  
  }  
}
```

Description: The passenger type code has been modified.

With mismatching passenger travel document gender:

```
json   
{  
  "messages": [  
    {  
      "rawValue": "Passenger (Doe, John) Gender doesn't match Travel Document (US, P, P123412341234) Gender",  
      "code": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",  
      "type": "FailedTravelDocValidation",  
      "value": null,  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": {  
    "oldPassengerKey": "MCFBRFQ-",  
    "newPassengerKey": "MCFTT1I-"  
  }  
}
```

Description: The passenger gender does not match with the travel document, and the `syncPassengerAndTravelDocGender` is set to false.

GET /api/nsk/v2/booking/passengers/{passengerKey}/seats

Summary: Gets all seats associated with a passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Passengers to get seats for.

Responses

200 Response

Ok.

404 Response

DELETE /api/nsk/v1/booking/passengers/{passengerKey}/seats/{unitKey}

Summary: Deletes a seat assignment from a specific passenger on the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	No description
unitKey	path	string	Yes	The unique unit key.
WaiveFee	query	boolean	No	The flag indicating to waive the seat fees if permissions permit.

```
plaintext
|
|
```

| IgnoreSeatSsrs | query | boolean | No | Flag indicating whether to ignore seat SSR's. |

Responses

200 Response

No content if successful.

GET /api/nsk/v2/booking/passengers/{passengerKey}/seats/{unitKey}

Summary: Gets passenger seat by key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Unique passenger key.
unitKey	path	string	Yes	Unique unit key.

Responses

200 Response

Ok.

404 Response

Not found.

POST /api/nsk/v2/booking/passengers/{passengerKey}/seats/{unitKey}

Summary: Adds a seat assignment to a specific passenger on the booking in state.

Description

This endpoint actively checks whether the passenger has a seat. If the passenger does have a seat, it will delete the current seat the passenger occupies and add the requested one. This may result in slightly slower response times.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
unitKey	path	string	Yes	The unit key.

Request Body

The add seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/seats/{unitKey}/quote

Summary: Retrieves a seat fee quote for a specific passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
unitKey	path	string	Yes	The unit key.
InventoryControlType	query	string	No	The type of hold placed on seats.

Enumeration values: 0 = Session, 1 = None || IgnoreSeatSSRs | query | boolean | No | Determines whether or not SSRs associated with the seat will be ignored when determining unit availability. || SeatAssignmentMode | query | string | No | The seat assignment mode can affect available units in seat maps and whether or not seat fees are charged based on channel type, role settings, customer programs and/or class of service.

Enumeration values: 0 = AutoDetermine, 1 = PreSeatAssignment, 2 = WebCheckin || MonetaryOnly | query | boolean | No | Indicates whether the fee to be priced is to be priced in money or points. || CollectedCurrencyCode | query | string | No | Optional currency code. If different than the booking currency code, fees in that currency will be returned if the conversion exists in the system. |

Responses

200 Response

400 Response

404 Response

Examples

Response 200

Retrieve Seat Fee Quote Response:

```
json   
{  
    "data": {  
        "type": 9,  
        "ssrCode": null,  
        "ssrNumber": 0,  
        "paymentNumber": 0,  
        "isConfirmed": false,  
        "isConfirming": true,  
        "isConfirmingExternal": false,  
        "passengerKey": "MCFBRFQ-",  
        "code": "JSF",  
        "detail": null,  
        "passengerFeeKey": "MCEwITk-",  
        "override": false,  
        "flightReference": "20241111 NV5670 SLCBOS",  
        "note": null,  
        "createdDate": "2025-08-07T04:47:16.4286445Z",  
        "isProtected": false,  
        "serviceCharges": [  
            {  
                "amount": 25.0,  
                "code": "JSF",  
                "detail": null,  
                "type": 6,  
                "collectType": 0,  
                "currencyCode": "USD",  
                "foreignCurrencyCode": "PHP",  
                "foreignAmount": 1435.0,  
                "ticketCode": "JSF"  
            }  
        ]  
    }  
}
```

Description: Successful retrieval of the seat fee information of the specified passenger.

Response 400

With No Booking in State:

```
json   
{  
    "errors": [  
        {  
            "id": "505b55d3-cac8-93a9-4d25-131256cdd551",  
            "code": "nsk:Booking:NoBookingInState",  
            "message": "nsk:Booking:NoBookingInState",  
            "type": "Validation",  
            "details": null,  
            "rawMessage": "No booking was found in session, cannot perform operation.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        }  
    ],  
    "data": null  
}
```

Description: No booking was put into state before retrieving a seat fee quote.

With Invalid Passenger Key:

© 2024 MaaS API. All rights reserved. This document is subject to change without notice.

```
json
{
  "errors": [
    {
      "id": "2987bed7-5690-5050-55e9-819b8551b344",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'PassengerKey' with value 'MCFBASDFRFQ-' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: The passenger key does not exist in the booking in state.

With Invalid Unit Key:

```
json
{
  "errors": [
    {
      "id": "2987bed7-5690-5050-55e9-819b8551b344",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'unitKey' with value 'MS3NIP5DHg09' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: The unit key is invalid and cannot be decoded properly.

With Unconfigured Collected Currency Code:

```
json
{
  "errors": [
    {
      "id": "62be6a4f-507b-a109-e082-2a2a1dd6a63e",
      "code": "nsk-server:CurrencyConversionRateNotFound",
      "message": "nsk-server:CurrencyConversionRateNotFound",
      "type": "Information",
      "details": null,
      "rawMessage": "No currency conversion rate found from USD to YER. You need to add this rate in Management Application.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: The collected currency code does not exist in the system or no conversion rate is found for the collected currency code.

With Invalid Collected Currency Code Length:

```

json
{
  "errors": [
    {
      "id": "169290f3-341a-c49e-f84c-c8049929b79f",
      "code": "SeatFeeQuoteRequestv2.CollectedCurrencyCode:StringLengthAttribute",
      "message": "The field SeatFeeQuoteRequestv2 must be a string with a maximum length of 3.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "SeatFeeQuoteRequestv2",
        "member": "CollectedCurrencyCode"
      },
      "rawMessage": "The field SeatFeeQuoteRequestv2 must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: The collected currency code exceeds the maximum length.

GET /api/nsk/v1/booking/passengers/{passengerKey}/ssrs/{ssrCode}/price

Summary: Gets an SSR price quote on the booking in state.

Description

Requires a booking in state.

GraphQL endpoints: passengerSsrPriceQuotes, passengerSsrPriceQuotesBatch

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
ssrCode	path	string	Yes	The ssr code.
collectedCurrencyCode	query	string	No	The collected currency code.

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/{passengerKey}/voucher

Summary: Gets information regarding a specific voucher code and passenger. Returns what a voucher can pay for a given passenger.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	The passenger key.
ReferenceCode	query	string	No	The voucher reference code.

plaintext



| OverrideRestrictions | query | boolean | No | Override restrictions. Defaults to false. |

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/passengers/breakdown

Summary: Gets all the passenger price breakdowns.

Description

No description available

Request Details

Responses

200 Response

The collection of passenger breakdowns.

404 Response

GET /api/nsk/v1/booking/passengers/breakdown/byType

Summary: Gets all the passenger price breakdown by passenger type.

Description

No description available

Request Details

Responses

200 Response

The collection of passenger type breakdowns.

404 Response

PUT /api/nsk/v1/booking/passengers/groupBooking

Summary: Updates the collection of un-named or TBA passengers on the booking.

Description

When Enforce Standard Group Booking setting is true, group name is required and un-named or TBA passengers will be setup with First Name: TBA, Middle Name: Sequential Numeric, Last Name = Group Name. When the setting is false, group name is not required and passengers will be setup with First Name: Sequential Alphabetic, Last Name: TBA

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

The collection of passengers.

500 Response

Booking/Payments Endpoints

GET /api/nsk/v1/booking/payments

Summary: Retrieves the booking payments on the booking in state.

Description

Requires a booking in state.

Request Details

Responses

200 Response

OK

404 Response

POST /api/nsk/v5/booking/payments

Summary: Creates a new external payment or prepaid payment on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext

The payment fields are dynamic to the criteria of your payment provider. Here is an example request for a credit card payment that does not require address, zip, etc.

```
{ "amount": 10, "currencyCode": "USD", "paymentFields": {"ACCTNO": "411111111111", "EXPDATE": "10/8/2017", "CC::AccountHolderName": "Bob Smith", "CC::VerificationCode": "111"}}
```

A 202 (HTTP status code) response returns alternate data signifying that either Three D Secure was enabled for this payment and is required for this payment type or a direct currency conversion (Dcc) offer was given. The payment will need to be resent using the 3DS endpoint or the Dcc endpoint.

Note that it is possible for a payment to return a dcc offer and also have three d secure enabled. In these cases the direct currency conversion offer will need to be handled first and then the three d secure payment can be handled.

If you would like to make a Dcc offer and the system using this endpoint cannot display a dcc offer to the end user, the status should be updated to DccNotOffered. This indicates that the offer was not displayed to the end user and is to be accepted automatically.

If you would like to make a ThreeDSecure payment, you will need to provide a valid term url. You will also need to have these headers populated:

User Agent (USER-AGENT)

Accept (ACCEPT)

This endpoint can also only be used with the v6 three d secure endpoint, POST `api/nsk/v6/booking/payments/threeDSecure` .

Request Details

Parameters

Name	Location	Type	Required	Description
termUrl	query	string	No	The term URL.

Request Body

Payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The booking payments response.

202 Response

The booking payments response.

400 Response

Examples

Response 202

Three D Secure 1 Redirect needed:

```
json
{
  "threeDSecureResponse": {
    "redirectUrl": "https://merchantacsstag.cardinalcommerce.com/MerchantACSWeb/pareq.jsp?vaa=b&gold=AAAAAAAAAAAAAAAAAAAAAA",
    "paReq": "eNpdkl1PwjAUhu/7K4jxeu26gUoOTUAiHxGyyILgXe10YMZ90G2A/952bKL26jyn57TnfVsI9pxvEJVaRSwwKKQ0+zE0eBGrTchn7JgVgSvz7v3X03m",
    "responseUrl": null,
    "md": null,
    "raw": "<html><head></head><body><form id=\"redirectionPostForm\" action=\"https://merchantacsstag.cardinalcommerce.com/Mercha",
    "paymentTransactionKey": "Y1ZYVDJIMFBJc1BXTGdicGNnSjAhMSEtMSEtMSFYWFhYWFnYWFnYWFnMDA3",
    "threeDSecureVersion": "1.0"
  },
  "directCurrencyConversionOffer": null
}
```

Description: Example response for a Three D Secure 1 redirect is required.

DCC Offer:

```
json
{
  "threeDSecureResponse": null,
  "directCurrencyConversionOffer": {
    "dccPaymentTransactionKey": "MSFYWFhYWFnYWFnYWFnODA0LjAwITAwMDAwMDAwLTAwMDAtMDAwMC0wMDAwLTAwMDAwMDAwMDAwMA--",
    "currencyCode": "AUD",
    "rateValue": 2.0,
    "amount": 2804.0
  }
}
```

Description: Example response for when a DCC Offer is proposed.

Response 400

Paying with a Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is None."
      },
      "rawMessage": "Blacklisted Card, the reason given is None.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card without a reason.

Paying with a Lost Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Lost."
      },
      "rawMessage": "Blacklisted Card, the reason given is Lost.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as lost.

Paying with a Stolen Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Stolen."
      },
      "rawMessage": "Blacklisted Card, the reason given is Stolen.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as stolen.

Paying with a Chargedback Blacklisted Card:

```

json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:ValidationException:BlackListedCard",
      "message": "nsk-server:Payment:ValidationException:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is ChargedBack."
      },
      "rawMessage": "Blacklisted Card, the reason given is ChargedBack.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when attempting to pay with a blacklisted card tagged as chargedback.

GET /api/nsk/v1/booking/payments/{paymentKey}

Summary: Gets a specific payment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	Payment key.

Responses

200 Response

Metadata response.

404 Response

DELETE /api/nsk/v1/booking/payments/{paymentKey}

Summary: Deletes the payment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	Payment key.

Responses

200 Response

Metadata response.

400 Response

PATCH /api/nsk/v1/booking/payments/{paymentKey}/authorize

Summary: Manually authorize a committed payment.

Description

NOTE: This endpoint takes a singular property for the request body and as such the request will need to be formatted differently than what it typically looks like. For more information, please see <https://navitaire-developer-digital-api.azurewebsites.net/articles/Miscellaneous/UniqueInputSingleRequestBodiesREST> and the Swagger request examples. This endpoint is to be used with caution and is only permissible to use under certain circumstances.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	The payment key.
force	query	boolean	No	Indicates if a manual force authorization is desired. Defaults to the standard, manual
authorization process if this query parameter is not provided. On a manual force authorization,				
only pending, amounted payments (under an external payment method) are allowed, and the agent's role				
must allow the manual force authorization of pending payments.				

plaintext



|

Request Body

The authorization code.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Examples

Request

Manual Authorize JSON request body, with an authorization code:

json



"ac89c98fdaj1dsfa"

Description: Example request for when you want to authorize a payment.

PATCH /api/nsk/v1/booking/payments/{paymentKey}/decline

Summary: Manually decline a committed payment.

Description

This endpoint is to be used with caution and is only permissible to use under certain circumstances.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	The payment key.
force	query	boolean	No	Indicates if a manual force decline is desired. Defaults to the standard, manual decline process if
this query parameter is not provided. On a manual force decline, only pending, amounted payments (under an external payment method) are allowed, and the agent's role must allow the manual force decline of pending payments.				

plaintext



Responses

200 Response

POST /api/nsk/v1/booking/payments/{paymentMethod}/dcc

Summary: Gets the available direct currency codes offer. This is for making an inline DCC payment.

Description

This endpoint behaves like a GET but is masked as a POST. If a DCC offer is accepted the payment transaction will need to be completed before getting another DCC offer. For example, a user cannot get a DCC offer for a VISA, get a DCC offer for a MasterCard and then accept the VISA DCC offer and pay with the VISA. The only way to complete this would be to get a DCC offer on one card, accept, and then pay and then get the DCC offer for another card, accept, and pay. Regardless of whether the offer is accepted or not, the endpoint /api/nsk/v6/booking/payments/dcc/{dccKey} will need to be called.

Request Details

Parameters

Name	Location	Type	Required	Description
paymentMethod	path	string	Yes	The payment method code.

Request Body

The DCC availability request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v1/booking/payments/allocations

Summary: Gets the payment allocations for the booking in state. Payment allocations are a collection of the payments assigned to each passenger on the booking.

Description

Request Details

Responses

200 Response

404 Response

GET /api/nsk/v2/booking/payments/available

Summary: Retrieves the booking payment methods available for the booking in state.

Description

Requires a booking in state.

Request Details

Parameters

Name	Location	Type	Required	Description
currencyCode	query	string	No	The optional currency code. Defaults to the logged in users currency code.

Responses

200 Response

OK

404 Response

POST /api/nsk/v1/booking/payments/blacklist

Summary: Adds the booking payment to the blacklist.

Description

Request Details

Request Body

The blacklist payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PATCH /api/nsk/v1/booking/payments/blacklist/{paymentKey}

Summary: Modifies the specified blacklisted payment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	The unique identifier of the blacklisted payment to modify.

Request Body

The modify blacklisted payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

404 Response

The item was not found.

DELETE /api/nsk/v1/booking/payments/blacklist/{paymentKey}

Summary: Removes the specified payment from the blacklist.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	The unique identifier of the payment to be removed from the blacklist.

Responses

200 Response

GET /api/nsk/v2/booking/payments/blacklist/{paymentKey}

Summary: Retrieves the specified blacklisted payment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
paymentKey	path	string	Yes	The unique identifier of the blacklisted payment to retrieve.

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

404 Response

The item was not found.

Examples

Response 200

Retrieve Blacklisted Payment:

```
json   
{  
  "data": {  
    "reason": 2,  
    "notes": "The credit card was lost",  
    "startDate": "2025-08-08T04:47:16.2299786Z",  
    "endDate": "2025-08-12T04:47:16.229979Z",  
    "paymentKey": "MzU5NTcxITEhTUMhMzg1MjAy",  
    "createdDate": "2025-08-07T04:47:16.2299778Z",  
    "modifiedDate": "2025-08-10T04:47:16.229978Z"  
  }  
}
```

Description: Successful retrieval of the blacklisted payment.

Response 400

Payment Not Found:

```
json   
{  
  "errors": [  
    {  
      "code": "nsk:Payment:PaymentKeyNotFoundOnBooking",  
      "message": "The payment key FdZvq0FzTaNkJ8M6dKtU is not found on the booking in state.",  
      "type": "Validation",  
      "number": null,  
      "details": {  
        "paymentKey": "FdZvq0FzTaNkJ8M6dKtU"  
      }  
    }  
  ]  
}
```

Description: Payment key does not exist in the booking in state

No Booking in State:

```
json   
{  
  "errors": [  
    {  
      "code": "nsk:Booking:NoBookingInState",  
      "message": "No booking was found in session, cannot perform operation.",  
      "type": "Validation",  
      "number": null,  
      "details": null  
    }  
  ]  
}
```

Description: No booking was put into state before retrieving a blacklisted payment.

GET /api/nsk/v2/booking/payments/bookingCredit

Summary: Gets credit available from a past booking.

Description

See booking retrieve. This method uses the same validation rules.

```
plaintext   
  
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
CurrencyCode	query	string	No	The currency code of the account.

```
plaintext
```

| RecordLocator | query | string | No | Record locator being requested. || EmailAddress | query | string | No | The email address of a contact. || Origin | query | string | No | The origin station code. || FirstName | query | string | No | The first name of a passenger or contact. || LastName | query | string | No | The last name of a passenger or contact. || CustomerNumber | query | string | No | One of the passengers customer numbers applied. || DepartureDate | query | string (date-time) | No | The first journeys departure date. |

Responses

200 Response

400 Response

404 Response

Examples

Response 400

Booking Search Failed:

```
json
```

```
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:GeneralRestriction",
      "message": null,
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when booking is not found, either from an invalid PNR or strategy configuration.

POST /api/nsk/v3/booking/payments/bookingCredit

Summary: Applies credit from a past booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Apply booking credit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Metadata.

400 Response

Examples

Response 400

Booking Search Failed:

```
json Copy

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:GeneralRestriction",
      "message": null,
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Example response when booking is not found, either from an invalid PNR or strategy configuration.

POST /api/nsk/v3/booking/payments/credit

Summary: Applies credit by reference number and type.

Description

This endpoint is for agents and can apply credit for any user or type.

```
plaintext Copy

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Request Body

Apply credit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v2/booking/payments/credit

Summary: Gets credit available by reference number and type.

Description

This endpoint is for agents and can get credit available for any user or type.

Request Details

Parameters

Name	Location	Type	Required	Description
ReferenceNumber	query	string	No	The account reference. This will be the record locator, customer account number, or organization code depending on the type.

```
plaintext
|
|
```

| CurrencyCode | query | string | No | The currency code of the account. || Type | query | string | No | Defines the type of credit being applied.

Enumeration values: 0 = Customer, 1 = Booking, 2 = Organization |

Responses

200 Response

Credit account HTTP response.

404 Response

POST /api/nsk/v3/booking/payments/customerCredit

Summary: Applies credit from the logged in customer to the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Apply credit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Metadata.

GET /api/nsk/v2/booking/payments/customerCredit

Summary: Gets credit available for the logged in user on the booking in state.

Description

Depending on configuration, credit may be validated by the contacts and passengers on the booking.

Request Details

Parameters

Name	Location	Type	Required	Description
CurrencyCode	query	string	No	The currency code of the account.

```
plaintext
|
|
```

Responses

200 Response

Credit account http response.

404 Response

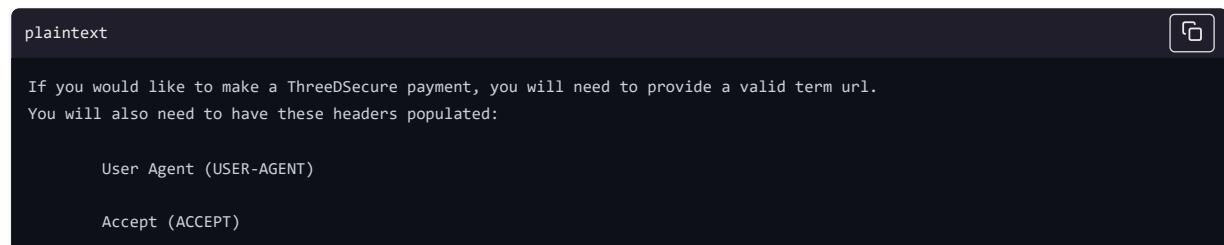
POST /api/nsk/v6/booking/payments/dcc/{dccKey}

Summary: Creates a new external payment with direct currency conversion. This endpoint is used for inline DCC payments only.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

If a DCC offer is accepted the payment transaction will need to be completed before getting another DCC offer. For example, a user cannot get a DCC offer for a VISA, get a DCC offer for a MasterCard and then accept the VISA DCC offer and pay with the VISA. The only way to complete this would be to get a DCC offer on one card, accept, and then pay and then get the DCC offer for another card, accept, and pay. Regardless of whether the offer is accepted or not, the endpoint /api/nsk/v5/booking/payments/dcc/{dccKey} will need to be called.



Request Details

Parameters

Name	Location	Type	Required	Description
dccKey	path	string	Yes	The DCC key.
termUrl	query	string	No	The term url.

Request Body

The DCC request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

400 Response

Examples

Response 400

Paying with a Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is None."
      },
      "rawMessage": "Blacklisted Card, the reason given is None.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card without a reason.

Paying with a Lost Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Lost."
      },
      "rawMessage": "Blacklisted Card, the reason given is Lost.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as lost.

Paying with a Stolen Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Stolen."
      },
      "rawMessage": "Blacklisted Card, the reason given is Stolen.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as stolen.

Paying with a Chargedback Blacklisted Card:

```

json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:Validation_Error:BlackListedCard",
      "message": "nsk-server:Payment:Validation_Error:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is ChargedBack."
      },
      "rawMessage": "Blacklisted Card, the reason given is ChargedBack.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when attempting to pay with a blacklisted card tagged as chargedback.

POST /api/nsk/v1/booking/payments/dcc/{dccPaymentTransactionKey}

Summary: Adds a dcc payment to the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

NOTE: This endpoint takes a singular property for the request body and as such the request will need to be formatted differently than what it typically looks like. For more information, please see <https://navitaire-developer-digital-api.azurewebsites.net/articles/Miscellaneous/UniqueInputSingleRequestBodiesREST> and the Swagger request examples. If the system using this endpoint cannot display a dcc offer to the end user, the status should be updated to DccNotOffered. This indicates that the offer was not displayed to the end user and is it to be accepted automatically.

```

plaintext

A 202 (HTTP status code) response returns alternate data signifying that Three D Secure was enabled for this payment and is required for this payment type. The payment will need to be resent using the 3DS endpoint.

If you would like to make a ThreeDSecure payment, you will need to provide a valid term url. You will also need to have these headers populated:

User Agent (USER-AGENT)

Accept (ACCEPT)

```

Request Details

Parameters

Name	Location	Type	Required	Description
dccPaymentTransactionKey	path	string	Yes	The dcc payment transaction key.

Request Body

The dcc status.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

404 Response

Examples

Request

Accepting a Dcc Offer JSON request body, 1:

```
json  
2
```

Description: Example request for when wanting to accept a dcc offer.

Accepting a Dcc Offer JSON request body, 2:

```
json  
"DccOfferAccepted"
```

Description: Example request for when wanting to accept a dcc offer.

POST /api/nsk/v1/booking/payments/fees/{feeCode}

Summary: Retrieves the payment fee for the fee code by providing the account number and payment method code for a manual form of payment.

Description

This endpoint behaves like a GET endpoint but is masked as POST.

Request Details

Parameters

Name	Location	Type	Required	Description
feeCode	path	string	Yes	The feeCode.

Request Body

The payment fee price request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

404 Response

GET /api/nsk/v1/booking/payments/fees/{feeCode}/storedPayment/{storedPaymentKey}

Summary: Retrieves the payment fee for the fee code by providing the stored payment key for a stored payment. The stored payment key will be able to provide to retrieve the account number, payment method code and account number id.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
feeCode	path	string	Yes	The fee code.
storedPaymentKey	path	string	Yes	The stored payment key of the agent holding the session.
Amount	query	number (decimal)	No	The amount being payed.

```
plaintext 
```

|

| CurrencyCode | query | string | No | The currency code. This will default to the bookings currency code. || CollectedCurrencyCode | query | string | No | The optional collected currency code. This will be needed for MCC requests. |

Responses

200 Response

404 Response

DELETE /api/nsk/v1/booking/payments/inProcess

Summary: Cancels an in-process payment, if one exists, from the session state. This can be used when in the middle of a 3DS, MCC, or DCC payment to remove the payment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Responses

200 Response

GET /api/nsk/v3/booking/payments/mcc

Summary: Gets the available multi-currency codes available.

Description

This is affected by the booking currency code.

Request Details

Responses

200 Response

404 Response

POST /api/nsk/v6/booking/payments/mcc/{currencyCode}

Summary: Creates a new MCC payment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

This is affected by the booking currency code. The MCC payment cannot have the same currency code in the URL and body. In a normal MCC rate scenario, the rate ID and exchange rate payment fields are auto-populated and should not be passed in manually.

plaintext



If you would like to make a ThreeDSecure payment, you will need to provide a valid term url.
You will also need to have these headers populated:

User Agent

Accept

Request Details

Parameters

Name	Location	Type	Required	Description
currencyCode	path	string	Yes	The collected currency code.
termUrl	query	string	No	The term URL.

Request Body

The payment method request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

400 Response

Examples

Request

Normal MCC Payment Request:

json



```
{  
    "thirdPartyExternalRate": null,  
    "paymentMethodCode": "MC",  
    "amount": 5.0,  
    "deposit": null,  
    "paymentFields": {  
        "acctno": "5210000010001001",  
        "cC::AccountHolderName": "Jim Bob",  
        "cC::VerificationCode": "123"  
    },  
    "waiveFees": false,  
    "currencyCode": null,  
    "installments": null  
}
```

Description: Example request for making a normal MCC payment.

Third Party External Rate MCC Payment Request:

```
json
{
  "thirdPartyExternalRate": {
    "rateId": "v2I0fc",
    "quotedCurrency": "USD",
    "collectedCurrency": "AUD",
    "exchangeRate": 1.31
  },
  "paymentMethodCode": "MC",
  "amount": 5.0,
  "deposit": null,
  "paymentFields": {
    "acctno": "5210000010001001",
    "cC::AccountHolderName": "Jim Bob",
    "cC::VerificationCode": "123"
  },
  "waiveFees": null,
  "currencyCode": null,
  "installments": null
}
```

Description: NOT RECOMMENDED TO USE. Example request for making an Mcc payment with a third party external rate.

Response 400

Paying with a Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is None."
      },
      "rawMessage": "Blacklisted Card, the reason given is None.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card without a reason.

Paying with a Lost Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Lost."
      },
      "rawMessage": "Blacklisted Card, the reason given is Lost.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as lost.

Paying with a Stolen Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Stolen."
      },
      "rawMessage": "Blacklisted Card, the reason given is Stolen.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as stolen.

Paying with a Chargedback Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is ChargedBack."
      },
      "rawMessage": "Blacklisted Card, the reason given is ChargedBack.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as chargedback.

POST /api/nsk/v6/booking/payments/mcc/{currencyCode}/storedPayment/{storedPaymentKey}

Summary: Creates a new MCC payment from a stored payment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

This is affected by the booking currency code. The MCC payment cannot have the same currency code in the URL and body. In a normal MCC rate scenario, the rate ID and exchange rate payment fields are auto-populated and should not be passed in manually.

```
plaintext

If you would like to make a ThreeDSecure payment, you will need to provide a valid term url.
You will also need to have these headers populated:

User Agent
Accept
```

Request Details

Parameters

Name	Location	Type	Required	Description
currencyCode	path	string	Yes	The collected currency code.
storedPaymentKey	path	string	Yes	Stored payment key.
termUrl	query	string	No	The term URL.

Request Body

The payment method request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

400 Response

Examples

Request

Normal MCC Stored Payment Request:

```
json   
{  
    "thirdPartyExternalRate": null,  
    "amount": 5.0,  
    "paymentFields": null,  
    "currencyCode": null,  
    "installments": null  
}
```

Description: Example request for making a normal MCC stored payment.

Third Party External Rate MCC Stored Payment Request:

```
json   
{  
    "thirdPartyExternalRate": {  
        "rateId": "v2I0fc",  
        "quotedCurrency": "USD",  
        "collectedCurrency": "AUD",  
        "exchangeRate": 1.31  
    },  
    "amount": 5.0,  
    "paymentFields": null,  
    "currencyCode": null,  
    "installments": null  
}
```

Description: NOT RECOMMENDED TO USE. Example request for making an MCC stored payment with a third party external rate.

Response 400

Paying with a Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is None."
      },
      "rawMessage": "Blacklisted Card, the reason given is None.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card without a reason.

Paying with a Lost Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Lost."
      },
      "rawMessage": "Blacklisted Card, the reason given is Lost.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as lost.

Paying with a Stolen Blacklisted Card:

```
json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Stolen."
      },
      "rawMessage": "Blacklisted Card, the reason given is Stolen.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as stolen.

Paying with a Chargedback Blacklisted Card:

```

json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationException:BlackListedCard",
      "message": "nsk-server:Payment:ValidationException:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is ChargedBack."
      },
      "rawMessage": "Blacklisted Card, the reason given is ChargedBack.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when attempting to pay with a blacklisted card tagged as chargedback.

POST /api/nsk/v3/booking/payments/organizationCredit

Summary: Applies credit to the logged in user's organization. NOTE: It is not recommended to use a custom agency credit payment as it will not have the same behavior nor be validated like the pre-configured AG credit payment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Apply credit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v2/booking/payments/organizationCredit

Summary: Gets credit available for the logged in users organization.

Description

Depending on configuration, credit may be validated by the contacts and passengers on the booking.

Request Details

Parameters

Name	Location	Type	Required	Description
CurrencyCode	query	string	No	The currency code of the account.

```

plaintext
| 

```

Responses

200 Response

Credit account http response.

404 Response

GET /api/nsk/v1/booking/payments/refunds

Summary: Retrieves the booking payment methods available for a refund on the booking in state.

Description

Requires a booking in state.

Request Details

Responses

200 Response

OK

404 Response

POST /api/nsk/v5/booking/payments/refunds

Summary: Creates a refund for the booking in state. No longer handles Credit Shell or CS refunds.

Description

Requires a booking in state. NOTE: CS, credit shell refunds are no longer accepted through this endpoint. It is now recommended to use POST `api/nsk/v1/booking/payments/refunds/creditShell` endpoint for any credit shell refund.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v1/booking/payments/refunds/creditShell

Summary: Creates a credit shell refund for the booking in state.

Description

Requires a booking in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The credit shell refund request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created

400 Response

Bad request.

Examples

Response 400

No Booking In State:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Booking:NoBookingInState",  
      "message": "No booking was found in session, cannot perform operation.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example of no booking in state.

Invalid Credit Shell Account Type:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Payment:InvalidCreditShellAccountType",  
      "message": "The payment credit shell account type CustomerNumber is not valid for the current user.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example of anonymous user attempting to create a customer credit.

Invalid Payment Method Code For Refund:

json

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Payment:PaymentMethodCodeInvalidForRefund",  
      "message": "The payment method code CS is not allowed for refund.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example of sending an empty or invalid refund request.

POST /api/nsk/v3/booking/payments/refunds/customerCredit

Summary: Creates a customer credit for a specified customer.

Description

Requires a booking in state and an agent token. Agents should use /refunds endpoint for booking credits.

plaintext

```
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
skipBalanceValidation	query	boolean	No	Indicates whether the validation of the booking balance should be skipped.

Request Body

The customer credit refund request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v2/booking/payments/refunds/organizationCredit

Summary: Creates an organization account refund for a payment from the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The organization refund request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST /api/nsk/v2/booking/payments/reversals

Summary: Reverses a payment for the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

This endpoint works similarly with the refund endpoint (POST api/nsk/v4/booking/payments/refunds), but agent-only permissions are needed and does not require a zero or negative booking balance for processing. Requires a booking in state. NOTE: CS, credit shell refunds are no longer accepted through this endpoint. It is now recommended to use POST api/nsk/v1/booking/payments/refunds/creditShell endpoint for any credit shell refund.

Request Details

Request Body

The payment reversal request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

409 Response

Concurrent changes were being made in the same session.

POST /api/nsk/v7/booking/payments/storedPayment/{storedPaymentKey}

Summary: Creates a payment using the stored payment information for the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext



Account Number and Expiration Date are stored.
With stored payments, payment fields may still be required to complete the transaction.
For example, a credit card that requires address information, this information is not stored with the stored payment.

A 202 (HTTP status code) response returns alternate data signifying that either Three D Secure was enabled for this payment and is required for this payment type or a direct currency conversion (Dcc) offer was given. The payment will need to be resent using the 3DS endpoint or the Dcc endpoint.
Note that it is possible for a payment to return a dcc offer and also have three d secure enabled. In these cases the direct currency conversion offer will need to be handled first and then the three d secure payment can be handled.

If you would like to make a Dcc offer and the system using this endpoint cannot display a dcc offer to the end user, the status should be updated to DccNotOffered. This indicates that the offer was not displayed to the end user
and is to be accepted automatically.

If you would like to make a ThreeDSecure payment, you will need to provide a valid term url.
You also need to have these headers populated:

User Agent (USER-AGENT)

Accept (ACCEPT)

Request Details

Parameters

Name	Location	Type	Required	Description
storedPaymentKey	path	string	Yes	Stored payment key.
termUrl	query	string	No	The term url.

Request Body

Payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

400 Response

Examples

Response 400

Paying with a Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is None."
      },
      "rawMessage": "Blacklisted Card, the reason given is None.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card without a reason.

Paying with a Lost Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Lost."
      },
      "rawMessage": "Blacklisted Card, the reason given is Lost.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as lost.

Paying with a Stolen Blacklisted Card:

```
json
{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217ccb8",
      "code": "nsk-server:Payment:ValidationError:BlackListedCard",
      "message": "nsk-server:Payment:ValidationError:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is Stolen."
      },
      "rawMessage": "Blacklisted Card, the reason given is Stolen.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when attempting to pay with a blacklisted card tagged as stolen.

Paying with a Chargedback Blacklisted Card:

```

json

{
  "errors": [
    {
      "id": "23c0d2aa-9c93-9c37-f9e4-39276217cbb8",
      "code": "nsk-server:Payment:Validation_Error:BlackListedCard",
      "message": "nsk-server:Payment:Validation_Error:BlackListedCard",
      "type": "Validation",
      "details": {
        "blacklistErrorMessage": "Blacklisted Card, the reason given is ChargedBack."
      },
      "rawMessage": "Blacklisted Card, the reason given is ChargedBack.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when attempting to pay with a blacklisted card tagged as chargedback.

POST /api/nsk/v6/booking/payments/threeDSecure

Summary: Creates a new ThreeDSecure external payment or prepaid payment on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

```

plaintext

A call to payments post should be made first and processed to verify a 3DS payment is required.

If you would like to make a ThreeDSecure payment, you will need to provide a valid term url.
You will also need to have these headers populated:

User Agent (USER-AGENT)

Accept (ACCEPT)

```

Request Details

Parameters

Name	Location	Type	Required	Description
termUrl	query	string	No	The term URL.

Request Body

Payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The booking payments response.

Examples

Request

Request:

```

json

{
  "paRes": "eNrdWVmPq0iyfrfk/3DU8+jpZjM2jHxKShZjbBazL2/sBoNZbQy//qarzllD94zUc19GUiVZBU1kZETGF/FFws68dEnCGU1075K3nZz0fZA1X/L462+Cpf
  "paReq": "eNpdkk1TwjAQhu/5FQzjuW1SEIZZMgPiKAeBgQq0t5ju0Cr9IG35+vUmpRU1p302u8m+bwJ+qBEnK1S1RgEvmOdyi60oGLafXmfvU/3pr8rLfbN6matj1h
  "paymentTransactionKey": "R1V0WklyalRTdXpPV2gzcDBybTEhMSEtMSEtMSFYWFhYWFlhYWFgMDA3"
}

```

Description: Example request for Three D Secure v6 POST.

POST /api/nsk/v4/booking/payments/voucher

Summary: Creates a new voucher payment on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created

GET /api/nsk/v3/booking/payments/voucher

Summary: Gets information regarding a specific voucher code and validates whether the referenced voucher exists as a voucher payment in the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ReferenceCode	query	string	No	The voucher reference code.

```

plaintext
| 

```

| OverrideRestrictions | query | boolean | No | Override restrictions. Defaults to false. |

Responses

200 Response

The requested voucher if available.

400 Response

403 Response

404 Response

Examples

Response 200

Valid Voucher Requested:

```
json   
{  
  "data": {  
    "calculatedAmount": 7.0,  
    "customerNumber": null,  
    "foreignCalculatedAmount": 0.0,  
    "foreignCalculatedCurrencyCode": null,  
    "foreignReversibleAmount": 0.0,  
    "nameRestriction": 0,  
    "redeemableAmount": 7.0,  
    "reversibleAmount": 0.0,  
    "foreignAvailableCurrencyCode": "",  
    "availableAmount": 1000.0,  
    "foreignCurrencyCode": null,  
    "foreignAmount": 1000.0,  
    "type": 1,  
    "password": null,  
    "recordLocator": null,  
    "transactions": null,  
    "voucherKey": "voucherKey",  
    "reference": "voucherReferenceNumber",  
    "configurationCode": "VOUCH",  
    "status": 0,  
    "available": 1000.0,  
    "firstName": null,  
    "lastName": null,  
    "personKey": null,  
    "voucherIssuanceKey": "voucherIssuanceKey",  
    "expiration": null,  
    "currencyCode": "USD",  
    "amount": 1000.0  
  }  
}
```

Description: Example Voucher Returned

All Fields Filled Voucher Response:

```
json 
```

```
{  
    "data": {  
        "calculatedAmount": 124.0,  
        "customerNumber": "customerNumber",  
        "foreignCalculatedAmount": 0.0,  
        "foreignCalculatedCurrencyCode": "",  
        "foreignReversibleAmount": 0.0,  
        "nameRestriction": 2,  
        "redeemableAmount": 0.0,  
        "reversibleAmount": 0.0,  
        "foreignAvailableCurrencyCode": "",  
        "availableAmount": 0.0,  
        "foreignCurrencyCode": "",  
        "foreignAmount": 0.0,  
        "type": 0,  
        "password": "NOTSUPPORTED",  
        "recordLocator": "XXXXXX",  
        "transactions": [  
            {  
                "voucherTransactionKey": "voucherTransactionKey",  
                "type": 1,  
                "recordLocator": "XXXXXX",  
                "foreignCurrencyCode": "",  
                "foreignAmount": 0.0,  
                "currencyCode": "USD",  
                "amount": 100.0  
            },  
            {  
                "voucherTransactionKey": "voucherTransactionKey1",  
                "type": 0,  
                "recordLocator": "XXXXXX",  
                "foreignCurrencyCode": "",  
                "foreignAmount": 0.0,  
                "currencyCode": "USD",  
                "amount": 100.0  
            }  
        ],  
        "voucherKey": "voucherKey",  
        "reference": "voucherReferenceNumber",  
        "configurationCode": "VOUCH",  
        "status": 2,  
        "available": 0.0,  
        "firstName": "firstName",  
        "lastName": "lastName",  
        "personKey": "personKey",  
        "voucherIssuanceKey": "voucherIssuanceKey",  
        "expiration": "2025-08-08T00:00:00+00:00",  
        "currencyCode": "USD",  
        "amount": 100.0  
    }  
}
```

Description: Example Voucher Returned

Response 400

Voucher Calculated Amount is 0:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Voucher:VoucherUnavailable",
      "message": "Requested Voucher has no Redeemable Balance. This can be due to any number of reasons including Balance already used or the Voucher has expired.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "messages": [
    {
      "code": "nsk:Voucher:VoucherValidationWarning",
      "type": "VoucherValidation",
      "value": "nsk:Voucher:VoucherValidationWarning",
      "status": 0,
      "details": {
        "voucherWarningMessage": "Journey: NV5610 Sat, 18Nov SLC-BOS 19:00-23:59 did not match any of the voucher market restrictions"
      }
    }
  ],
  "data": null
}

```

Description: Example error if a voucher has a calculated amount of 0, or doesn't belong to the requester.

Voucher Reference Not Supplied:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherRequest.ReferenceCode:RequiredAttribute",
      "message": "The VoucherRequest field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherRequest",
        "member": "ReferenceCode"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example error if no Voucher Reference is included on the request.

[DELETE /api/nsk/v2/booking/payments/voucher/{voucherPaymentReference}](#)

Summary: Reverses a voucher payment on the booking in state. This refunds the claimed amount on the voucher.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherPaymentReference	path	string	Yes	Voucher payment reference.

Responses

200 Response

The voucher deletion response.

POST /api/nsk/v1/booking/payments/voucher/passenger/{passengerKey}

Summary: Creates a new voucher payment for a specified passenger.

Description

For a passenger-select voucher to be shared across passengers, please use the standard voucher payment endpoint.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
passengerKey	path	string	Yes	Passenger key.

Request Body

Passenger voucher payment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

202 Response

Booking/Retrieve Endpoints

GET /api/nsk/v2/booking/retrieve

Summary: Retrieves the booking from the New Skies server based on the retrieve booking request parameters and any configured booking retrieve strategies. If the booking exists and the search criteria is allowed by one of the configured booking retrieve strategies, the booking will be loaded into the web session and returned.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

This endpoint will perform validation using a pre-configured booking retrieve strategy. Custom strategies can be configured in Utilities, and are configurable based on application name. To set-up a custom strategy, go to Utilities -> Resource Management. From the "Resource Management" screen, go to Root -> Web -> dotREZ folder. Right click on the dotREZ folder and select "Add Content". On the "Add Resource" screen, set the "Content Name" to "BookingStrategyManagementConfig". Choose "XML" as the "Type". The "Content Data" will be in JSON, and should look like this:

```
plaintext

{
  "strategies" :
  [
    {
      "applicationName" : "dotREZApi",
      "allowedStrategies" : [
        "firstAndLastName",
        "email"
      ]
    },
    {
      "applicationName" : "Web",
      "allowedStrategies" : [
        "email"
      ]
    },
    {
      "applicationName" : "",
      "allowedStrategies" : [
        "lastName"
      ]
    }
  ]
}
```

The application name can be set when creating a new token. If no application name is given for a token create request, the API will use whatever value is defined in the "dotrez-nsk-common.json" configuration file under the "systemOptions" -> "fallbackApplicationName". If no strategies are configured and no "fallbackApplicationName" is configured, the default strategy will be a booking retrieve based on PNR, first name, and last name. If the same strategy should be used for multiple applications, the last entry in the example above can be set by setting an empty string as the application name, and then listing the allowed strategies to be used. Application specific strategies can still be defined, but if no application specific strategy is found, a list of strategies associated with the blank application name will be used (if one exists in the configuration file). The list of strategies currently includes the following options:

```
plaintext

  firstAndLastName

  email

  originAndDepartureDate

  lastName

  customerNumber
```

More strategies may be added in the future.

Request Details

Parameters

Name	Location	Type	Required	Description
RecordLocator	query	string	No	Record locator being requested.

```
plaintext

|
```

| EmailAddress | query | string | No | The email address of a contact. || Origin | query | string | No | The origin station code. || FirstName | query | string | No | The first name of a passenger or contact. || LastName | query | string | No | The last name of a passenger or contact. || CustomerNumber | query | string | No | One of the passengers customer numbers applied. || DepartureDate | query | string (date-time) | No | The first journeys departure date. |

Responses

200 Response

400 Response

404 Response

Examples

Response 400

Booking Search Failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:GeneralRestriction",
      "message": null,
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when booking is not found, either from an invalid PNR or strategy configuration.

GET /api/nsk/v1/booking/retrieve/{bookingKey}

Summary: Gets a specific booking by key and stores it in state.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Description

This action requires a session token and the user logged in.

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Responses

200 Response

OK. Request completed successfully.

401 Response

Authorization Not Allowed.

404 Response

The item was not found.

409 Response

Concurrent changes were being made in the same session.

GET /api/nsk/v1/booking/retrieve/byRecordLocator/{recordLocator}

Summary: Gets a specific booking by record locator and stores it in state.

Description

This action requires a session token and the user logged in.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking record locator.

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

401 Response

Authorization Not Allowed.

404 Response

The item was not found.

409 Response

Concurrent changes were being made in the same session.

Examples

Response 200

Retrieve booking by record locator:

json



```
{
  "data": {
    "selfServiceMoveAvailable": false,
    "bookingKey": "MTU4NjExTVBDNlo0SCFmYWxzzQ--",
    "recordLocator": "PC6Z4H",
    "currencyCode": "USD",
    "systemCode": "SYS",
    "groupName": "GRP",
    "locators": [
      {
        "numericRecordLocator": "310530162615",
        "parentRecordLocator": "PB5L2S",
        "parentId": 0,
        "recordLocators": [
          {
            "recordLocatorKey": "NjA4NjQyMjchTlY-",
            "recordCode": "60864227",
            "systemDomainCode": "MVS",
            "owningSystemCode": "NV",
            "bookingSystemCode": "NV",
            "interactionPurpose": "ML",
            "hostedCarrierCode": "MK"
          }
        ]
      }
    ],
    "info": {
      "status": 2,
      "paidStatus": 1,
      "priceStatus": 2,
      "profileStatus": 1,
      "bookingType": "Default",
      "channelType": 1,
      "bookedDate": "2025-08-07T04:47:15.6026872Z",
      "createdDate": "2025-08-07T04:47:15.602688Z",
      "expirationDate": "2025-08-10T04:47:15.6026881Z",
      "modifiedDate": "2025-08-07T04:50:15.6026891Z",
      "modifiedAgentId": 1,
      "createdAgentId": 1,
      "owningCarrierCode": "NV",
      "changeAllowed": true,
      "createdUserKey": "MQ--",
      "modifiedUserKey": "MQ--"
    },
    "sales": {
      "created": {
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWW"
      },
      "source": {
        "isoCountryCode": "LA",
        "sourceSystemCode": "LA",
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWW"
      },
      "modified": {
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWW"
      }
    },
    "typeOfSale": {
      "residentCountry": "US",
      "promotionCode": "PROMO",
      "fareTypes": [
        "FareType"
      ]
    }
  }
}
```

```
"hold": {
    "expiration": "2025-08-10T04:47:15.6026958Z"
},
"breakdown": {
    "balanceDue": 0.0,
    "pointsBalanceDue": 0.0,
    "authorizedBalanceDue": 0.0,
    "totalAmount": 66.44,
    "totalPoints": 0.0,
    "totalToCollect": 66.44,
    "totalPointsToCollect": 0.0,
    "totalCharged": 66.44,
    "passengerTotals": {
        "services": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        },
        "specialServices": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        },
        "seats": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        },
        "upgrades": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [

```

```
{
    "amount": 25.0,
    "code": "JSF",
    "detail": "SLC-BOS",
    "type": 23,
    "collectType": 0,
    "currencyCode": "USD",
    "foreignCurrencyCode": "USD",
    "foreignAmount": 25.0,
    "ticketCode": "JSF"
}
]
},
"spoilage": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"nameChanges": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"convenience": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"infant": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [

```

```
{
    "amount": 25.0,
    "code": "JSF",
    "detail": "SLC-BOS",
    "type": 23,
    "collectType": 0,
    "currencyCode": "USD",
    "foreignCurrencyCode": "USD",
    "foreignAmount": 25.0,
    "ticketCode": "JSF"
}
]
},
"totalCost": null
},
"passengers": {
    "mcfbrfQ-": {
        "passengerKey": "MCFBRFQ-",
        "services": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        },
        "specialServices": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        },
        "seats": {
            "total": 25.0,
            "taxes": 0.0,
            "adjustments": 0.0,
            "charges": [
                {
                    "amount": 25.0,
                    "code": "JSF",
                    "detail": "SLC-BOS",
                    "type": 23,
                    "collectType": 0,
                    "currencyCode": "USD",
                    "foreignCurrencyCode": "USD",
                    "foreignAmount": 25.0,
                    "ticketCode": "JSF"
                }
            ]
        }
    }
},
```

```
"upgrades": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"spoilage": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"nameChanges": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
"convenience": {
    "total": 25.0,
    "taxes": 0.0,
    "adjustments": 0.0,
    "charges": [
        {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
        }
    ]
},
```

```

    "infant": {
      "total": 25.0,
      "taxes": 0.0,
      "adjustments": 0.0,
      "charges": [
        {
          "amount": 0.3,
          "code": "INFT",
          "detail": "Infant Fee",
          "type": 23,
          "collectType": 0,
          "currencyCode": "USD",
          "foreignCurrencyCode": "USD",
          "foreignAmount": 0.3,
          "ticketCode": "INF"
        }
      ]
    },
    "totalCost": null
  },
  "journeyTotals": {
    "totalAmount": 40.0,
    "totalPoints": 0.0,
    "totalTax": 2.44,
    "totalDiscount": 0.0
  },
  "journeys": {
    "tlZ_NTY3MH4gfn5TTEN_MDEvMzEvMjAyNSAxNzozMH5CT1N_MDEvMzEvMjAyNSAyMzozMH5_": {
      "journeyKey": "tlZ_NTY3MH4gfn5TTEN_MDEvMzEvMjAyNSAxNzozMH5CT1N_MDEvMzEvMjAyNSAyMzozMH5_",
      "totalAmount": 40.0,
      "totalPoints": 0.0,
      "totalTax": 2.44,
      "totalDiscount": 0.0
    }
  },
  "addOnTotals": {
    "car": 1.0,
    "hotel": 0.0,
    "activities": 0.0
  },
  "receivedBy": {
    "receivedBy": "Doe, John",
    "latestReceivedBy": "Doe, John",
    "receivedReference": "DOE",
    "latestReceivedReference": "DOE",
    "referralCode": "REFERRAL"
  },
  "contacts": {
    "p": {
      "contactTypeCode": "P",
      "phoneNumbers": [
        {
          "type": 3,
          "number": "98123124123"
        }
      ],
      "cultureCode": "USD",
      "address": {
        "lineOne": "Address Line One",
        "lineTwo": "Address Line Two",
        "lineThree": "Address Line Three",
        "countryCode": "US",
        "provinceState": "UT",
        "city": "Salt Lake City",
        "postalCode": "84101"
      },
      "emailAddress": "johndoe@gmail.com",
      "customerNumber": "2050001326",
      "sourceOrganization": "WWW",
      "distributionOption": 0
    }
  }
}

```

```
"notificationPreference": 0,
"companyName": "Company Name",
"name": {
  "first": "John",
  "middle": "Cee",
  "last": "Doe",
  "title": "MR",
  "suffix": null
},
},
"passengers": {
  "mcfbrfq-": {
    "passengerKey": "MCFBRFQ-",
    "passengerAlternateKey": "MTk2MjE1",
    "customerNumber": "2050001326",
    "fees": [
      {
        "type": 9,
        "ssrCode": "SSR",
        "ssrNumber": 365,
        "paymentNumber": 0,
        "isConfirmed": false,
        "isConfirming": false,
        "isConfirmingExternal": false,
        "code": "JSF",
        "detail": "Fee Details",
        "passengerFeeKey": "MCEwITk-",
        "override": false,
        "flightReference": "20250131 NV5670 SLCBOS",
        "note": "Fee Notes",
        "createdDate": "2025-08-07T04:47:15.602741Z",
        "isProtected": false,
        "serviceCharges": [
          {
            "amount": 25.0,
            "code": "JSF",
            "detail": "SLC-BOS",
            "type": 23,
            "collectType": 0,
            "currencyCode": "USD",
            "foreignCurrencyCode": "USD",
            "foreignAmount": 25.0,
            "ticketCode": "JSF"
          }
        ]
      }
    ],
    "name": {
      "first": "John",
      "middle": "Cee",
      "last": "Doe",
      "title": "MR",
      "suffix": null
    },
    "passengerTypeCode": "ADT",
    "discountCode": "ADT",
    "bags": [
      {
        "identifier": "5670",
        "baggageKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
        "nonStandard": false,
        "type": "Backpack",
        "osTag": "NV3665",
        "osTagDate": "2025-08-07T04:47:15.6027439Z",
        "taggedToStation": "BOS",
        "stationCode": "SLC",
        "weight": 5,
        "taggedToCarrierCode": "JanSport Backpack",
        "weightType": 2
      }
    ],
  }
},
```

```
"program": {
    "code": "ADT",
    "levelCode": "LVL",
    "number": "365"
},
"infant": {
    "fees": [],
    "nationality": "US",
    "dateOfBirth": "2024-08-07T04:47:15.6027492+00:00",
    "travelDocuments": [
        {
            "passengerTravelDocumentKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
            "documentTypeCode": "PASS",
            "birthCountry": "USA",
            "issuedByCode": "USA",
            "name": {
                "first": "John",
                "middle": "Cee",
                "last": "Doe",
                "title": "MR",
                "suffix": "JR"
            },
            "nationality": "US",
            "expirationDate": "2035-08-07T04:47:15.6027463Z",
            "number": "P31322423218",
            "issuedDate": "2024-08-07T04:47:15.6027478Z",
            "gender": 1,
            "dateOfBirth": "2024-08-07T04:47:15.6027482Z",
            "declaredGender": "M",
            "placeOfBirth": "Salt Lake City",
            "placeOfIssue": "Salt Lake City"
        }
    ],
    "residentCountry": "US",
    "gender": 1,
    "name": {
        "first": "John",
        "middle": "Cee",
        "last": "Doe",
        "title": "MR",
        "suffix": "JR"
    },
    "type": "INFT"
},
"info": {
    "nationality": "US",
    "residentCountry": "US",
    "gender": 1,
    "dateOfBirth": "2001-08-07T04:47:15.6027587Z",
    "familyNumber": 1
},
"travelDocuments": [
    {
        "passengerTravelDocumentKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
        "documentTypeCode": "PASS",
        "birthCountry": "USA",
        "issuedByCode": "USA",
        "name": {
            "first": "John",
            "middle": "Cee",
            "last": "Doe",
            "title": "MR",
            "suffix": null
        },
        "nationality": "US",
        "expirationDate": "2035-08-07T04:47:15.6027513Z",
        "number": "P31322423218",
        "issuedDate": "2015-08-07T04:47:15.6027514Z",
        "gender": 1,
        "dateOfBirth": "2001-08-07T04:47:15.6027516Z",
        "declaredGender": "M",
        "placeOfBirth": "Salt Lake City",
        "placeOfIssue": "Salt Lake City"
    }
]
```

```
        "placeOfIssue": "Salt Lake City"
    },
],
"addresses": [
{
    "status": 0,
    "companyName": "Company Name",
    "lineOne": "Line One",
    "passengerAddressKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
    "phone": "801-555-1212",
    "lineTwo": "Line Two",
    "stationCode": "SLC",
    "lineThree": "Line Three",
    "emailAddress": "johndoe@gmail.com",
    "countryCode": "US",
    "cultureCode": "US",
    "provinceState": "UT",
    "refusedContact": false,
    "city": "Salt Lake City",
    "postalCode": "84101"
},
],
"weightCategory": 1,
"emdCoupons": [
{
    "status": 0,
    "passengerEmdCouponKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
    "emdTypeCode": 0,
    "emdNumber": "1234567890123",
    "couponNumber": 1,
    "optionalServiceInformation": {
        "carrierCode": "NV",
        "reasonForIssuanceSubCode": "RFISC"
    },
    "salesDate": "2025-08-07T04:47:15.6027551Z",
    "flightReference": "20250131 NV5670 SLCBOS",
    "receivedFromSystemCode": "GDS",
    "numberOfServices": 1,
    "feeType": 9,
    "feeKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
    "ssrCode": "SSR",
    "ssrNumber": 365
}
],
}
],
"journeys": [
{
    "flightType": 1,
    "stops": 0,
    "designator": {
        "destination": "BOS",
        "origin": "SLC",
        "arrival": "2025-08-08T04:47:15.6028053Z",
        "departure": "2025-08-07T04:47:15.6028054Z"
    },
    "move": {
        "maxMoveBackDays": 365,
        "maxMoveOutDays": 365
    },
    "segments": [
{
        "isStandby": false,
        "isConfirming": false,
        "isConfirmingExternal": false,
        "isBlocked": false,
        "isHosted": true,
        "isChangeOfGauge": false,
        "designator": {
            "destination": "BOS",
            "origin": "SLC",
            "arrival": "2025-08-08T04:47:15.6028053Z",
            "departure": "2025-08-07T04:47:15.6028054Z"
        }
    }
]
```

```
"departure": "2025-08-07T04:47:15.6028054Z",
},
"isSeatmapViewable": true,
"fares": [
{
  "isGoverning": true,
  "carrierCode": "NV",
  "fareKey": "MH5Fn50Vn5FU1BUU0ZSRX55UFRTfn4wfjB_flg-",
  "classOfService": "E",
  "classType": "Class Type",
  "fareApplicationType": 0,
  "fareClassOfService": "E",
  "fareBasisCode": "ERPTSFRE",
  "fareSequence": 0,
  "inboundOutBound": 0,
  "fareStatus": 0,
  "isAllotmentMarketFare": false,
  "originalClassOfService": "E",
  "ruleNumber": "RPTS",
  "productClass": "C2",
  "ruleTariff": "RPTS",
  "travelClassCode": "C",
  "pricingDate": "2025-08-07T04:47:15.6027667Z",
  "crossReferenceClassOfService": "E",
  "fareDesignator": {
    "fareTypeCodes": [
      "A",
      "AP",
      "B",
      "C"
    ]
  },
  "passengerFares": [
    {
      "serviceCharges": [
        {
          "amount": 2.44,
          "code": "USTAX",
          "detail": "SLC-BOS",
          "type": 5,
          "collectType": 0,
          "currencyCode": "USD",
          "foreignCurrencyCode": "USD",
          "foreignAmount": 2.44,
          "ticketCode": "US"
        }
      ],
      "discountCode": "ADT",
      "fareDiscountCode": "20DDISC",
      "passengerType": "ADT"
    }
  ],
  "fareLink": 0
}
],
"segmentKey": "T1Z_NTY3MH4gfn5TTEN_MDEvMzEvMjAyNSAxNzozMH5CT1N_MDEvMzEvMjAyNSAyMzozMH5_",
"identifier": {
  "identifier": "5670",
  "carrierCode": "NV",
  "opSuffix": "N"
},
"passengerSegment": {
  "mcnbrfQ": {
    "seats": [
      {
        "compartmentDesignator": "C",
        "penalty": 0,
        "unitDesignator": "7G",
        "seatInformation": {
          "deck": 1,
          "seatSet": 1,
          "propertyList": {
            "id": "1234567890"
          }
        }
      }
    ]
  }
}
```

```
        "aisle": "TRUE",
        "bulkhead": "TRUE",
        "tcc": "C"
    }
},
"arrivalStation": "BOS",
"departureStation": "SLC",
"passengerKey": "MCFBRFQ-",
"unitKey": "T1YhNTY3MCEgITYz0Dcz0TQxNDAwMDAwMCFTTEMhQk9TITdHIUM-",
"crossReferenceSeatingPreference": "A",
"isPending": false,
"seatmapReference": "T1YhNTY3MCEgITYz0Dcz0TQxNDAwMDAwMCFTTEMhQk9T"
}
],
"passengerKey": "MCFBRFQ-",
"activityDate": "2025-08-07T04:47:15.6027687Z",
"boardingSequence": "0",
"createdDate": "2025-08-07T04:47:15.6027692Z",
"liftStatus": 0,
"modifiedDate": "2025-08-07T04:57:15.6027695Z",
"overBookIndicator": 0,
"priorityDate": "2025-08-07T04:47:15.6027698Z",
"timeChanged": false,
"verifiedTravelDocs": null,
"sourcePointOfSale": {
    "agentCode": "Agent",
    "domainCode": "TAW3",
    "locationCode": "TAW3",
    "organizationCode": "WWW"
},
"pointOfSale": {
    "agentCode": "Agent",
    "domainCode": "TAW3",
    "locationCode": "TAW3",
    "organizationCode": "WWW"
},
"ssrs": [
{
    "isConfirmed": false,
    "isConfirmingUnheld": false,
    "note": "Note",
    "ssrDuration": 2,
    "ssrKey": "MTU4NjExIVBDNlo0SCFmYWxzzQ--",
    "count": 1,
    "ssrCode": "SSR",
    "feeCode": "JSF",
    "inBundle": true,
    "passengerKey": "MCFBRFQ-",
    "ssrDetail": "Detail",
    "ssrNumber": 365,
    "market": {
        "identifier": {
            "identifier": "5670",
            "carrierCode": "NV",
            "opSuffix": "N"
        },
        "destination": "BOS",
        "origin": "SLC",
        "departureDate": "2025-08-07T04:47:15.6027787Z"
    }
}
],
"tickets": [
{
    "ticketNumber": "1234567890123",
    "infantTicketNumber": "1234567890124",
    "ticketIndicator": 1,
    "ticketStatus": 1,
    "passengerKey": "MCFBRFQ-"
}
],
"bags": [
```

```
{
    "baggageKey": "MTU4NjExIVBDNlo0SCFmYWxzZQ--",
    "passengerKey": "MCFBRFQ-",
    "arrivalStation": "BOS",
    "status": 0,
    "departureStation": "SLC",
    "osTag": "NV3665"
}
],
"scores": [
{
    "guestValueCode": "GVC",
    "score": 100,
    "passengerKey": "MCFBRFQ-"
}
],
"boardingPassDetail": {
    "gateInformation": "Gate Information",
    "priorityInformation": "Priority Information",
    "cabinClass": "Cabin Class",
    "compartmentLevel": "Compartment level",
    "boardingZone": "Boarding Zone",
    "seatAssignment": "Seat Assignment",
    "sequenceNumber": "Sequence Number"
},
"hasInfant": true,
"seatPreferences": {
    "seat": 2,
    "travelClass": 2,
    "advancedPreferences": [
        {
            "seatMapCode": "AISLE",
            "value": "TRUE",
            "status": 1
        }
    ]
},
"bundleCode": "CC02",
"verifiedTravelDocuments": [
    "MTU4NjExIVBDNlo0SCFmYWxzZQ--"
],
"referenceNumber": 35642,
"baggageGroupNumber": 2
},
},
"channelType": 1,
"cabinOfService": "C",
"externalIdentifier": {
    "identifier": "5660",
    "carrierCode": "MK",
    "opSuffix": "M"
},
"priorityCode": "R",
"changeReasonCode": 0,
"segmentType": 0,
"salesDate": "2025-08-07T04:47:15.602792Z",
"international": false,
"flightReference": "20250131 NV5670 SLCBOS",
"legs": [
{
    "legKey": "NjM4NzM5NDE0MDAwMDAwIUSWITU2NzAhICFTTEMhQk9TITI0MDYxMDQ-",
    "operationsInfo": {
        "arrivalGate": {
            "estimatedGate": "First Gate",
            "actualGate": "Second Gate"
        },
        "estimatedArrivalTime": "2025-08-07T07:47:15.6027955Z",
        "departureGate": {
            "estimatedGate": "First Gate",
            "actualGate": "Second Gate"
        },
        "actualOffBlockTime": "2025-08-08T04:47:15.6027979Z",
        "estimatedDepartureTime": "2025-08-08T07:47:15.6027955Z"
    }
}
]
```

```
"actualOnBlockTime": "2025-08-08T04:47:15.602798Z",
"actualTouchDownTime": "2025-08-09T04:47:15.6027981Z",
"airborneTime": "2025-08-08T07:47:15.6027982Z",
"arrivalNote": "Arrival Note",
"arrivalStatus": 0,
"baggageClaim": "Baggage Claim",
"departureNote": "Departure Note",
"departureStatus": 0,
"departureTimes": {
    "scheduled": "2025-08-07T04:47:15.6027989Z",
    "estimated": "2025-08-07T04:37:15.6027991Z"
},
"standardArrivalTime": "2025-08-08T04:47:15.6027992Z",
"tailNumber": "9V-SKA"
},
"designator": {
    "destination": "BOS",
    "origin": "SLC",
    "arrival": "2025-08-08T04:47:15.6028053Z",
    "departure": "2025-08-07T04:47:15.6028054Z"
},
"legInfo": {
    "departureTimeUtc": "2025-08-07T11:47:15Z",
    "arrivalTimeUtc": "2025-08-08T09:47:15Z",
    "adjustedCapacity": 168,
    "arrivalTerminal": "Concourse A",
    "arrivalTimeVariant": -300,
    "backMoveDays": 365,
    "capacity": 171,
    "changeOfDirection": false,
    "codeShareIndicator": 0,
    "departureTerminal": "Concourse B",
    "departureTimeVariant": -420,
    "equipmentType": "767",
    "equipmentTypeSuffix": "200",
    "eTicket": false,
    "irop": false,
    "lid": 168,
    "marketingCode": "DJM_3R",
    "marketingOverride": false,
    "operatedByText": "Operated By NV",
    "operatingCarrier": "NV",
    "operatingFlightNumber": "5670",
    "operatingOpSuffix": "N",
    "outMoveDays": 365,
    "arrivalTime": "2025-08-08T04:47:15.602801Z",
    "departureTime": "2025-08-07T04:47:15.6028011Z",
    "prbcCode": "B767EMDJ",
    "scheduleServiceType": "S",
    "sold": 6,
    "status": 0,
    "subjectToGovtApproval": false
},
"nests": [
{
    "adjustedCapacity": 168,
    "classNest": 8,
    "lid": 168,
    "travelClassCode": "C",
    "nestType": 0,
    "legClasses": [
{
    "classNest": 8,
    "classAllotted": 0,
    "classType": "C",
    "classAuthorizedUnits": 0,
    "classOfService": "C",
    "classRank": 0,
    "classSold": 0,
    "cnxSold": 0,
    "latestAdvancedReservation": 0,
    "status": 0
}
]
}
```

```

        "thruSold": 0
    }
}
],
"ssrs": [
{
    "available": 88,
    "ssrNestCode": "1BAG",
    "lid": 168,
    "sold": 100,
    "unitSold": 80
}
],
"seatmapReference": "T1YhNTY3MCEgITYzODczOTQxNDAwMDAwMCFTTEMhQk9T",
"flightReference": "20250131 NV5670 SLCBOS"
}
],
"status": 6
},
"journeyKey": "TlZ_NTY3MH4gfFn5TTEN_MDEvMzEvMjAyNSAxNzozMH5CT1N_MDEvMzEvMjAyNSAyMzozMH5_",
"notForGeneralUser": false
}
],
"comments": [
{
    "type": 0,
    "text": "Comment",
    "createdDate": "2025-08-07T04:47:15.6028084Z",
    "pointOfSale": {
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWN"
    },
    "commentKey": "MASDFASDXCFQ-"
}
],
"queues": [
{
    "code": "AQCD",
    "subCode": "AQBD",
    "name": "Automation Queue Smoke",
    "queueId": 121940,
    "passengerId": 196215,
    "watchlistId": 23,
    "note": "Booking with balance due of 67.44",
    "type": 15,
    "action": 0,
    "mode": 0,
    "flightReference": "20250131 NV5670 SLCBOS",
    "bookingQueueKey": "MTIxOTQwIUFRQ0Qh",
    "passengerAlternateKey": "MTk2MjE1",
    "watchListKey": "MjM-"
}
],
"history": [
{
    "historyCategory": 31,
    "details": "Assigned Seat",
    "event": 4,
    "pointOfSale": {
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWN"
    },
    "sourcePointOfSale": {
        "agentCode": "Agent",
        "domainCode": "TAW3",
        "locationCode": "TAW3",
        "organizationCode": "WWN"
    }
}
]
}
]
```

```

        "organizationCode": "WWW"
    },
    "receivedBy": "Doe, John",
    "receivedByReference": "315231",
    "createdDate": "2025-08-07T04:47:15.6028162Z"
}
],
"payments": [
{
    "paymentKey": "MTUwNTA5ITEhQ0EhMTU4NjEx",
    "code": "VI",
    "approvalDate": "2025-08-07T04:47:15.6028189Z",
    "details": {
        "accountNumberId": 0,
        "parentPaymentId": 0,
        "accountName": "John Doe",
        "accountNumber": "5110920006810092",
        "expirationDate": "2029-08-07T04:47:15.6028204Z",
        "text": "Payment",
        "installments": 0,
        "binRange": 511092,
        "fields": {
            "payment Field": "Payment Field Value"
        }
    },
    "amounts": {
        "amount": 67.44,
        "currencyCode": "USD",
        "collected": 67.44,
        "collectedCurrencyCode": "USD",
        "quoted": 67.44,
        "quotedCurrencyCode": "USD"
    },
    "authorizationCode": "AUTH",
    "authorizationStatus": 4,
    "fundedDate": "2025-08-07T04:47:15.6028271Z",
    "transactionCode": "TRANSACTION",
    "dcc": {
        "rateId": "11bb6fd9-10e3-4d42-9d54-829faccabeda",
        "currencyCode": "USD",
        "rateValue": 1.0,
        "amount": 67.44,
        "putInState": "STATE",
        "status": 2,
        "applicable": true
    },
    "threeDSecure": {
        "browserUserAgent": "TAW3",
        "browserAccept": "SYS",
        "remoteIpAddress": "192.168.1.1",
        "termUrl": "www.bank.com",
        "paReq": "Payload",
        "acsUrl": "www.bank.com",
        "paRes": "Response",
        "authResult": "Approved",
        "cavv": "dsFL1IwiCigL2T4UqqKxOA==",
        "cavvAlgorithm": "Encryption",
        "eci": "05",
        "xid": "SDF235",
        "applicable": true,
        "successful": true,
        "threeDSecureSessionId": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJVbmtub3duIiwianRpIjoiNmYwNjhjYmQtNDBjYy05N2QyL",
        "challengePreference": 0,
        "challengeWindowSize": 0,
        "transactionIdThreeDSecure": "6f068cbd-40cc-97d2-e8bc-139591794461",
        "threeDSecureVersion": "1.0.2",
        "merchantData": "MerchantData",
        "redirectJwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJVbmtub3duIiwianRpIjoiMzNkZDg2ZjQtMDg40C0xMmQ4LTfiNmUtNwZ",
        "redirectJwtUrl": "www.bank.com",
        "dsTransactionId": "SDF235",
        "authenticationStatus": 1
    }
},

```

```
"attachments": [
  {
    "id": 365291,
    "paymentId": 3652391231,
    "attachment": "AQIDBAU="
  }
],
"createdDate": "2025-08-07T04:47:15.6028421Z",
"modifiedDate": "2025-08-07T04:52:15.6028422Z",
"type": 0,
"status": 3,
"transferred": false,
"channelType": 1,
"pointOfSale": {
  "agentCode": "Agent",
  "domainCode": "TAW3",
  "locationCode": "TAW3",
  "organizationCode": "WWW"
},
"sourcePointOfSale": {
  "agentCode": "Agent",
  "domainCode": "TAW3",
  "locationCode": "TAW3",
  "organizationCode": "WWW"
},
"deposit": false,
"accountId": 0,
"voucher": {
  "id": 0,
  "transactionId": 0,
  "overrideRestrictions": false,
  "overrideAmount": false,
  "recordLocator": "PC6Z4H"
},
"addedToState": true,
"createdAgentId": 1,
"modifiedAgentId": 1,
"reference": 150509,
"passengerVoucher": {
  "passengerVoucherKey": "NTA2MiExMTcwITEwMDExNzA3MDQzMzAwMDAx",
  "passengerKey": "MCFBRFQ-",
  "voucherConfigurationCode": "HNDACAR",
  "voucherUses": [
    13
  ],
  "customerPrograms": [
    {
      "programCode": "HNDA",
      "programNumber": "3215135"
    }
  ],
  "customerNumber": "2050001326"
}
},
"addOns": {
  "honda Civic": {
    "paymentRequired": false,
    "status": 0,
    "addOnKey": null,
    "type": 4,
    "summary": {
      "total": 100.0,
      "held": 0.0,
      "charged": 100.0,
      "supplierCode": "HONDA",
      "beginDate": "2025-08-07T04:47:15.6028652Z",
      "beginLocation": "SLC",
      "endDate": "2025-08-14T04:47:15.6028654Z",
      "endLocation": "BOS",
      "externalReference": "HONDA-123",
      "description": "Car Rental",
      "addOns": []
    }
  }
}]]
```

```
        "externalConfirmationNumber": "HDA365"
    },
    "reference": "Component Reference",
    "created": {
        "agentReference": "MTIxNTg-",
        "agentCode": "Agent",
        "date": "2025-08-07T04:47:15.6028682Z",
        "organizationCode": "WWW",
        "domainCode": "TAW3",
        "locationCode": "TAW3"
    },
    "source": {
        "agentCode": "Agent",
        "organizationCode": "WWW",
        "domainCode": "TAW3",
        "locationCode": "TAW3"
    },
    "declinedText": "Declined Component Text",
    "cultureCode": "US",
    "modifiedDate": "2025-08-07T05:47:15.6028693Z",
    "modifiedAgentReference": "Agent Reference",
    "order": {
        "address": {
            "coordinates": {
                "latitude": "40.7608",
                "longitude": "-111.951814"
            },
            "lineOne": "Address Line One",
            "lineTwo": "Address Line Two",
            "lineThree": "Address Line Three",
            "countryCode": "US",
            "provinceState": "UT",
            "city": "Salt Lake City",
            "postalCode": "84101"
        },
        "orderKey": "HNDA365",
        "phoneNumbers": [
            {
                "type": 3,
                "number": "98123124123"
            }
        ],
        "active": false,
        "locations": [
            {
                "description": "Salt Lake City Location",
                "code": "SLC",
                "utcOffset": 365.0,
                "usageDate": "0001-01-01T00:00:00"
            }
        ],
        "criteria": {
            "catalogCode": "A",
            "countryCode": "US",
            "supplierCode": "JPN",
            "departmentCode": "CARS",
            "vendorCode": "HNDA",
            "companyCode": "HNDA",
            "ratingCode": "10",
            "discountCode": "CVCDS",
            "promotionCode": "HNDA SALE",
            "currencyCode": "USD",
            "categoryCode": "CAR"
        },
        "customer": {
            "customerKey": "MCFBRFQ-",
            "name": {
                "first": "John",
                "middle": "Cee",
                "last": "Doe",
                "title": "MR",
                "suffix": null
            }
        }
    }
}
```

```
},
"customerNumber": "2050001326",
"address": {
    "county": "Salt Lake County",
    "lineOne": "Address Line One",
    "lineTwo": "Address Line Two",
    "lineThree": "Address Line Three",
    "countryCode": "US",
    "provinceState": "UT",
    "city": "Salt Lake City",
    "postalCode": "84101"
},
"dateOfBirth": "2001-08-07T04:47:15.602878Z",
"emailAddress": "johndoe@gmail.com",
"companyName": "Company Name",
"type": "Residential",
"homePhone": "833-81-21",
"workPhone": "801-555-1212",
"fax": "801-555-1212"
},
"thumbnailFileName": "File Name",
"participants": {
    "john Doe": {
        "participantKey": "MCFBRFO-",
        "description": "New Participant",
        "name": {
            "first": "John",
            "middle": "Cee",
            "last": "Doe",
            "title": "MR",
            "suffix": null
        },
        "participantTypeCode": "ADT",
        "address": {
            "county": "Salt Lake County",
            "lineOne": "Address Line One",
            "lineTwo": "Address Line Two",
            "lineThree": "Address Line Three",
            "countryCode": "US",
            "provinceState": "UT",
            "city": "Salt Lake City",
            "postalCode": "84101"
        },
        "isPrimary": true,
        "dateOfBirth": "2001-08-07T04:47:15.602882Z",
        "document": {
            "number": "1",
            "issuedByCode": "PRTICPANT",
            "documentTypeCode": "CHECK"
        },
        "emailAddress": "johndoe@gmail.com",
        "companyName": "Company Name",
        "type": "Residential",
        "homePhone": "833-81-21",
        "workPhone": "801-555-1212",
        "fax": "801-555-1212"
    }
},
"quantity": 10,
"history": [
    {
        "code": "Y",
        "previousCode": "N",
        "note": "Note",
        "created": "2025-08-07T04:47:15.602883Z",
        "hasError": false
    }
],
"usageDate": "0001-01-01T00:00:00",
"payment": {
    "paymentKey": "MTUwNTA5ITEhQ0EhMTU4NjEx",
    "type": "VI",
    "amount": 1000
}
```

```
"currencyCode": "USD",
"name": {
    "first": "John",
    "middle": "Cee",
    "last": "Doe",
    "title": "MR",
    "suffix": null
},
"number": "5110920006810092",
"expiration": "2029-08-07T04:47:15.6028855Z",
"cvv": "365",
"amount": 100.0,
"description": "Payment Description",
"issueNumber": "3656912",
"address": {
    "lineOne": "Address Line One",
    "lineTwo": "Address Line Two",
    "lineThree": "Address Line Three",
    "countryCode": "US",
    "provinceState": "UT",
    "city": "Salt Lake City",
    "postalCode": "84101"
},
"emailAddress": "johndoe@gmail.com",
"phoneNumbers": [
    {
        "type": 3,
        "number": "98123124123"
    }
],
"externalLocator": "JPN",
"parameters": [
    {
        "value": "10",
        "code": "HNDA",
        "description": "Parameter Description"
    }
],
"description": "Order Description",
"descriptionFormatType": "Description Format Type",
"productDescription": "Product Description",
"productVariationDescription": "Product Variation Description",
"paymentAction": 1,
"amounts": {
    "display": {
        "value": 25.0,
        "total": 50.0
    },
    "initial": {
        "value": 25.0,
        "total": 50.0
    },
    "markup": {
        "value": 25.0,
        "total": 50.0
    },
    "listed": {
        "value": 25.0,
        "total": 50.0
    },
    "listedDiscount": {
        "value": 25.0,
        "total": 50.0
    },
    "discount": {
        "value": 25.0,
        "total": 50.0
    },
    "handling": {
        "value": 25.0,
        "total": 50.0
    }
}
```

```
        },
        "handlingDiscount": {
            "value": 25.0,
            "total": 50.0
        },
        "dueNow": {
            "preTax": 25.0,
            "tax": 10.0,
            "total": 35.0
        },
        "dueLater": {
            "preTax": 25.0,
            "tax": 10.0,
            "total": 35.0
        },
        "personalizations": 10.0,
        "taxable": 35.0,
        "services": 50.0,
        "fees": 100.0,
        "total": 115.0,
        "taxRate": 2,
        "taxExempt": true,
        "taxAtUnitPrice": true
    },
    "terms": [
        {
            "code": "APPROVED",
            "description": "Description",
            "terms": "Terms"
        }
    ],
    "cancellationTerms": [
        {
            "code": "CANCELLED",
            "description": "Description",
            "terms": "Terms"
        }
    ],
    "details": [
        {
            "code": "HNDA",
            "styleCode": "H",
            "description": "Description"
        }
    ],
    "fees": [
        {
            "code": "HND",
            "feeCategoryCode": "VI",
            "description": "Fee Description",
            "amount": {
                "value": 25.0,
                "total": 50.0
            },
            "foreignAmount": {
                "value": 25.0,
                "total": 50.0
            },
            "type": 0,
            "isWaiveable": false,
            "foreignCurrencyCode": "USD",
            "currencyCode": "USD",
            "isChargeable": false
        }
    ],
    "notes": [
        {
            "text": "Order Text",
            "description": "Order Description",
            "created": "2025-08-07T04:47:15.6028969Z",
            "modified": "2025-08-07T05:17:15.602897Z"
        }
    ]
}
```

```

        ],
        "productVariationKey": "HNDA365",
        "productOrderKey": "HNDA365"
    },
    "isHistorical": false,
    "charges": [
        {
            "type": 6,
            "code": "SRVC",
            "ticketCode": "JSF",
            "collection": 0,
            "currencyCode": "USD",
            "amount": 64.0,
            "details": "Charge Details"
        }
    ]
},
"orders": [
{
    "bookingOrderKey": "HNDA365",
    "orderId": "ff0495a9-a48c-4b12-8bee-cf063e0f252a",
    "collect": 100.0,
    "thirdPartyCharge": 0.0,
    "thirdPartyHold": 0.0,
    "total": 100.0,
    "status": 1
}
]
}
}

```

Description: Successful retrieval of a booking using a record locator

Response 400

With Invalid Record Locator:

```

json
{
  "errors": [
    {
      "id": "96a5f7a8-184b-0284-7b76-37f03eba38bb",
      "code": "nsk:Booking:InvalidRecordLocator",
      "message": "The record locator AZJVGLAZJVGLAZJVGL is not valid.",
      "type": "Validation",
      "details": {
        "recordLocator": "AZJVGLAZJVGLAZJVGL"
      },
      "rawMessage": "The record locator AZJVGLAZJVGLAZJVGL is not valid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}

```

Description: The record locator provided is null, empty, or exceeded the maximum length.

With Non-Existent Booking:

```
json

{
  "errors": [
    {
      "id": "effe2fb4-0ff9-d384-37e6-47af652bdea3",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'recordLocator' with value 'AZJVGL' is invalid.",
      "type": "Validation",
      "details": {
        "recordLocator": "AZJVGL"
      },
      "rawMessage": "The identifier 'recordLocator' with value 'AZJVGL' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The booking associated with the record locator is not found.

With Authorization View Past Dated Booking Not Allowed:

```
json

{
  "errors": [
    {
      "id": "59d8f235-edcd-a6f1-f8ff-4935f1bacbd5",
      "code": "nsk-server:AuthorizationViewPastDatedBookingNotAllowed",
      "message": "Agent is not allowed to retrieve the booking.",
      "type": "Validation",
      "details": null,
      "rawMessage": "Agent is not allowed to retrieve the booking.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The agent is not allowed to retrieve a record locator of a past dated booking.

With Authorization Group Access Not Allowed:

```
json

{
  "errors": [
    {
      "id": "0e0ab88f-059f-d9a4-2c8d-843d8e2866c2",
      "code": "nsk-server:AuthorizationGroupAccessNotAllowed",
      "message": "Agent is not allowed to view/update the requested booking. Role Code: TAW3",
      "type": "Validation",
      "details": null,
      "rawMessage": "Agent is not allowed to view/update the requested booking. Role Code: TAW3",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The agent's group is not allowed to retrieve a booking.

GET /api/nsk/v1/booking/retrieve/byThirdPartyRecordLocator

Summary: Gets a specific booking by third party record locator and stores it in state.

Description

This action requires a session token and the user logged in.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
SystemCode	query	string	No	System Code being requested.

plaintext



|

| ThirdPartyRecordLocator | query | string | No | The Third Party Record Locator of the booking being requested. |

Responses

200 Response

OK. Request completed successfully.

401 Response

Authorization Not Allowed.

404 Response

The item was not found.

409 Response

Concurrent changes were being made in the same session.

Booking/Seatmaps Endpoints

GET /api/nsk/v3/booking/seatmaps

Summary: Gets the list of seat maps for all the journeys for the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
FeePricingMode	query	string	No	Fee pricing mode.

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly || CollectedCurrencyCode | query | string | No | The collected currency code. || IncludePropertyLookup | query | boolean | No | Flag indicating to include seat property lookup on the response. || CultureCode | query | string | No | The desired culture code. |

Responses

200 Response

The list of available seat maps.

400 Response

GET /api/nsk/v4/booking/seatmaps/journey/{journeyKey}

Summary: Gets the list of seat maps for an entire journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The unique journey key.
FeePricingMode	query	string	No	Fee pricing mode.

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly || CollectedCurrencyCode | query | string | No | The collected currency code. || IncludePropertyLookup | query | boolean | No | Flag indicating to include seat property lookup on the response. || CultureCode | query | string | No | The desired culture code. |

Responses

200 Response

The list of seat maps for the journey.

400 Response

GET /api/nsk/v3/booking/seatmaps/segment/{segmentKey}

Summary: Gets the list of seat maps for a specific journey's segment.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.
FeePricingMode	query	string	No	Fee pricing mode.

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly || CollectedCurrencyCode | query | string | No | The collected currency code. || IncludePropertyLookup | query | boolean | No | Flag indicating to include seat property lookup on the response. || CultureCode | query | string | No | The desired culture code. |

Responses

200 Response

The list of seat maps.

400 Response

Booking/Seats Endpoints

POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}

Summary: Auto assigns seats to all selected passengers that do not have them for every journey on the booking. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
primaryPassengerKey	path	string	Yes	The key for the primary passenger.

Request Body

The auto assign seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Seats auto-assigned.

POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/journey/{journeyKey}

Summary: Auto assigns seats to all passengers that do not have them for a specific journey. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
primaryPassengerKey	path	string	Yes	The key for the primary passenger.
journeyKey	path	string	Yes	The specific journey key to auto assign.

Request Body

The auto assign seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Seats auto-assigned.

POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/journey/{journeyKey}/quote

Summary: Retrieves the seat fee information of the auto-assigned seats of a passenger on a given journey.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The key of the journey for the auto-assigned seat where the seat fee information will be based.

plaintext 

| primaryPassengerKey | path | string | Yes | The key of the primary passenger being seated. |

Request Body

The seat fee quote request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/segment/{segmentKey}

Summary: Auto assigns seats to all passengers that do not have them for a specific journey's segment. If seatedNearPrimary is empty, all passengers in the state booking will be requested for seat assignments. Otherwise, passengers with a passenger key specified in seatedNearPrimary will be requested instead.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
primaryPassengerKey	path	string	Yes	The key for the primary passenger.
segmentKey	path	string	Yes	The specific segment key to auto assign.

Request Body

The auto assign seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Seats auto-assigned.

POST /api/nsk/v1/booking/seats/auto/{primaryPassengerKey}/segment/{segmentKey}/quote

Summary: Retrieves the seat fee information of the auto-assigned seats of a passenger on a given segment.

Description

Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The key of the segment for the auto-assigned seat where the seat fee information will be based.

plaintext
| primaryPassengerKey | path | string | Yes | The key of the primary passenger being seated. |

Request Body

The seat fee quote request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

Booking/Segments Endpoints

DELETE /api/nsk/v1/booking/segments/{segmentKey}

Summary: Cancels the specific segment from the booking in state.

Description

Calling this method will alter the existing journeyKey. If you are cancelling the only segment in a journey, this method will behave the same as calling DELETE on /api/nsk/v1/booking/journeys/{journeyKey}. If there are segments remaining after the delete, the Location response header will be set with the location of the modified journey with its new key.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The key of the specific segment to be canceled.

Responses

200 Response

DELETE /api/nsk/v1/booking/segments/{segmentKey}/classOfService

Summary: Restores a segment to the class of service prior to an upgrade or downgrade.

Description

This does not truly delete the old class of service. Instead, it restores the original class of service before an upgrade or downgrade occurred.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.
overSell	query	boolean	No	Indicates the intent to restore the class of service even if there are no available AUs.

Responses

200 Response

GET /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}

Summary: Gets a specific passenger segment from the booking in state.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.
passengerKey	path	string	Yes	The passenger key.

Responses

404 Response

200 Response

POST /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/infant

Summary: Creates an infant on a passenger in the specified segment.

Description

If the booking in state contains a passive segment, an infant will not be created on the passive segment for the passenger. If the passenger on the specified segment already has an infant, an exception will be thrown. If any other segment in the booking already has an infant, the infant details from the request will be ignored and the infant details from the existing infant will be retained.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The unique segment key.
passengerKey	path	string	Yes	The unique passenger key.

Request Body

The new infant.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

DELETE /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/infant

Summary: Deletes an infant from a passenger in the specified segment.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The unique identifier of the segment to remove infants from.
passengerKey	path	string	Yes	The unique passenger key.

Responses

200 Response

POST /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/tickets

Summary: Creates a ticket on a passenger segment on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.
passengerKey	path	string	Yes	The passenger key.

Request Body

Details to create the ticket.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT /api/nsk/v1/booking/segments/{segmentKey}/passengers/{passengerKey}/tickets

Summary: Updates a ticket on a passenger segment on the booking in state.

Description

If you attempt to update the passenger number and set the ticket indicator code to 'X' (TicketRequired) you will encounter an error. If you need to update both please do so separately.

```
plaintext 
```

This endpoint cannot be used to remove a ticket number, but can be used to update to a different ticket number.

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
segmentKey	path	string	Yes	The segment key.
passengerKey	path	string	Yes	The passenger key.

Request Body

Details to update the ticket.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PUT /api/nsk/v1/booking/segments/classOfService/{classModifyKey}

Summary: Modifies the class of service for the current booking in state to effect a segment upgrade or downgrade.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
classModifyKey	path	string	Yes	The class modify key from class of service availability.

Request Body

The class modify request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/nsk/v1/booking/segments/classOfService/availability

Summary: Gets class of service availability for the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
isUpgrade	query	boolean	No	Indicates an upgrade class of service request.
When set to false, indicates a downgrade class of service request.				

| collectedCurrencyCode | query | string | No | Collected currency code. || feePricingMode | query | string | No | Fee pricing mode. Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly || includeUnavailable | query | boolean | No | Indicates that classes of service with no availability should be returned. |

Responses

200 Response

404 Response

Booking/Ssrs Endpoints

GET /api/nsk/v1/booking/ssrs

Summary: Retrieves the current SSRs added to the booking in state.

Description

Requires a booking in state.

Request Details

Responses

200 Response

OK

404 Response

POST /api/nsk/v3/booking/ssrs

Summary: Adds SSRs to the booking in state.

Description

Adding an SSR by journey automatically adds the SSR to every segment. Requires a booking in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The SSR by keys request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

GET /api/nsk/v1/booking/ssrs/{ssrKey}

Summary: Retrieves the current SSR by key.

Description

Requires a booking in state.

Request Details

Parameters

Name	Location	Type	Required	Description
ssrKey	path	string	Yes	Unique Ssr passenger identifier.

Responses

200 Response

OK

404 Response

Not found.

PUT /api/nsk/v1/booking/ssrs/{ssrKey}

Summary: Modify the ssr note.

Description

NOTE: This endpoint takes a singular property for the request body and as such the request will need to be formatted differently than what it typically looks like. For more information, please see <https://navitaire-developer-digital-api.azurewebsites.net/articles/Miscellaneous/UniqueInputSingleRequestBodiesREST> and the Swagger request examples. Requires a booking in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
ssrKey	path	string	Yes	Unique Ssr key identifier.

Request Body

The updated note.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Examples

Request

Ssr Note, Json Request Body:

json



"Adding an ssr note."

Description: Example request for updating a ssr note.

DELETE /api/nsk/v1/booking/ssrs/{ssrKey}

Summary: Deletes by the SSR passenger key provided by metadata.

Description

Requires a booking in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
ssrKey	path	string	Yes	Unique Ssr passenger identifier.

Responses

200 Response

OK

POST /api/nsk/v3/booking/ssrs/{ssrKey}

Summary: Add an SSR to the booking in state.

Description

Adding an SSR by journey automatically adds the SSR to every segment. Requires a booking in state.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
ssrKey	path	string	Yes	The SSR key to add.

Request Body

The single SSR by key request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 201

With Already Booked SSR or Bundle:

```
json
{
  "messages": [
    {
      "rawValue": "[HMC] cannot be sold because passenger 0 already booked [THCS] on NV3075 YTZ-LAX.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "[HMC] cannot be sold because passenger 0 already booked [THCS] on NV3075 YTZ-LAX."
      }
    },
    "data": null
  }
}
```

Description: Example OSI Message if passenger already booked an SSR or bundle

With Cannot Find Interchangeable Bundle SSR:

```
json
{
  "messages": [
    {
      "rawValue": "Cannot find interchangeable bundle SSR for [HMC] in state.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "Cannot find interchangeable bundle SSR for [HMC] in state."
      }
    },
    "data": null
  }
}
```

Description: Example OSI Message if cannot find interchangeable bundle SSR in state

With SSR Not Found in Service Bundle:

```
json
{
  "messages": [
    {
      "rawValue": "SSR [HMC] not found in service bundle.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "SSR [HMC] not found in service bundle."
      }
    },
    "data": null
  }
}
```

Description: Example OSI Message if an SSR is not found in service bundle

With Duplicate SSR or Bundle:

```

json

{
  "messages": [
    {
      "rawValue": "Duplicate SSR [THCS] in SAN-YTZ is not allowed for service bundle.",
      "code": "CannotSellBundleSSR",
      "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are",
      "type": "CannotSellBundleSSR",
      "value": "CannotSellBundleSSR",
      "status": 0,
      "details": {
        "cannotSellBundleSSRMessage": "Duplicate SSR [THCS] in SAN-YTZ is not allowed for service bundle."
      }
    },
    "data": null
  }
}

```

Description: Example OSI Message if duplicate SSR or Bundle

POST /api/nsk/v2/booking/ssrs/availability

Summary: Get the ssr's available to add to the booking in state.

Description

- Requires a booking in state.
- By default, all ssr availability for the booking is returned.
- Infant SSR availability for the configured infant SSR is not included by default.
 - If the configured infant SSR availability needs to be displayed, the `passengerOptions.showInfantSsrAvailability` flag in the API's configuration should be set to true.
- Note that the SSR key for the default, configured infant SSR will be null as selling the infant SSR is handled via the following endpoints:
 - `booking/passengers/{passengerKey}/infant` POST/DELETE
 - `trip/sell` POST
 - `trip` POST

Request Details

Request Body

Optional request filter.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

POST /api/nsk/v1/booking/ssrs/bundles/resell

Summary: Resells any cancelled SSR bundle to the provided journey on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The bundle resell request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

DELETE /api/nsk/v1/booking/ssrs/manual

Summary: Deletes the SSR by the request criteria.

Description

Requires a booking in state. This is an alternate way of deleting an SSR. Deleting by ID is recommended (see /api/booking/ssrs). This method is useful if the UI is not querying availability and not using dynamic SSR information.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Unique Ssr passenger information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK

POST /api/nsk/v2/booking/ssrs/manual

Summary: Adds SSRs to the booking in state.

Description

Adding an SSR by journey automatically adds the SSR to every segment. Requires a booking in state. This is an alternate way of creating an SSR. Creating by ID is recommended (see /api/booking/ssrs). This method is useful if the UI is not querying availability and not using dynamic SSR information.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

403 Response

POST /api/nsk/v1/booking/ssrs/resell

Summary: Resells any cancelled ssrs to the provided journey on the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

The resell ssr request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The updated booking.

Bookings Endpoints

GET /api/nsk/v2/bookings

Summary: Retrieves the booking from the New Skies server based on the retrieve booking request parameters and any configured booking retrieve strategies. If the booking exists and the search criteria is allowed by one of the configured booking retrieve strategies, the booking will be returned, but not loaded into the web session.

Description

This endpoint will perform validation using a pre-configured booking retrieve strategy. Custom strategies can be configured in Utilities, and are configurable based on application name. To set-up a custom strategy, go to Utilities -> Resource Management. From the "Resource Management" screen, go to Root -> Web -> dotREZ folder. Right click on the dotREZ folder and select "Add Content". On the "Add Resource" screen, set the "Content Name" to "BookingStrategyManagementConfig". Choose "XML" as the "Type". The "Content Data" will be in JSON, and should look like this:

```
plaintext   
{  
  "strategies" :  
  [{  
    "applicationName" : "dotREZApi",  
    "allowedStrategies" : [  
      "firstAndLastName",  
      "email"  
    ]  
  }, {  
    "applicationName" : "Web",  
    "allowedStrategies" : [  
      "email"  
    ]  
  }, {  
    "applicationName" : "",  
    "allowedStrategies" : [  
      "lastName"  
    ]  
  }]  
}
```

The application name can be set when creating a new token. If no application name is given for a token create request, the API will use whatever value is defined in the "dotrez-nsk-common.json" configuration file under the "systemOptions" -> "fallbackApplicationName". If no strategies are configured and no "fallbackApplicationName" is configured, the default strategy will be a booking retrieve based on PNR, first name, and last name. If the same strategy should be used for multiple applications, the last entry in the example above can be set by setting an empty string as the application name, and then listing the allowed strategies to be used. Application specific strategies can still be defined, but if no application specific strategy is found, a list of strategies associated with the blank application name will be used (if one exists in the configuration file). The list of strategies currently includes the following options:

```
plaintext

firstAndLastName

email

originAndDepartureDate

lastName

customerNumber
```

More strategies may be added in the future.

Request Details

Parameters

Name	Location	Type	Required	Description
RecordLocator	query	string	No	Record locator being requested.

```
plaintext

|
```

| EmailAddress | query | string | No | The email address of a contact. || Origin | query | string | No | The origin station code. || FirstName | query | string | No | The first name of a passenger or contact. || LastName | query | string | No | The last name of a passenger or contact. || CustomerNumber | query | string | No | One of the passengers customer numbers applied. || DepartureDate | query | string (date-time) | No | The first journeys departure date. |

Responses

200 Response

400 Response

404 Response

Examples

Response 400

Booking Search Failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Booking:GeneralRestriction",
      "message": null,
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned." }
  ],
  "data": null
}
```

Description: Example response when booking is not found, either from an invalid PNR or strategy configuration.

GET /api/nsk/v1/bookings/{bookingKey}/history

Summary: Gets a itinerary history stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
Event	query	string	No	The event that triggered the history entry.

Enumeration values: 0 = Unknown, 1 = ConvertedHistory, 2 = FlightTimeChange, 3 = FlightDesignatorChange, 4 = AssignedSeat, 5 = RemoveSeat, 6 = AddedFlight, 7 = DeletedFlight, 8 = DeletedPassenger, 9 = NameChange, 10 = GroupNameChange, 11 = CancelledTicketing, 12 = ScheduleChange, 13 = AddedPayment, 14 = ServiceFee, 15 = QueuedPnr, 16 = UnqueuedPnr, 17 = DeletedComment, 18 = Divided, 19 = CheckedIn, 20 = CheckedOut, 21 = FareOverride, 22 = AddedBaggage, 23 = ChangedBaggageWeight, 24 = CheckedBaggage, 25 = RemovedBaggage, 26 = BoardedPassenger, 27 = UnboardedPassenger, 28 = ManualAuthorization, 29 = ManualDecline, 30 = UndoCancel, 31 = ItinerarySent, 32 = ContactChange, 33 = SsrAdded, 34 = FlightMoved, 35 = VerifiedDocument, 36 = RemovedVerifiedDocument, 37 = Promotion, 38 = BookingComment, 39 = CancelledSchedule, 40 = CancelServiceFee, 41 = OverrideServiceFee, 42 = AddedRecordLocator, 43 = DeletedRecordLocator, 44 = UpgradeClassOfService, 45 = DowngradeClassOfService, 46 = StandbyPriorityChange, 47 = AssignedTicketNumber, 48 = DeletedTicketNumber, 49 = ConfirmSegmentStatusCodeChange, 50 = CodeshareFlightChanged, 51 = PdsCancel, 52 = PdsPending, 53 = PdsConfirm, 54 = PdsFinalized, 55 = PdsDeclined, 56 = PdsException, 57 = PdsCancelRefused, 58 = PdsCancelUnsuccessful, 59 = Apps, 60 = InhibitedOverride, 61 = PrintedBagTag, 62 = SelfPrintedBagTag, 63 = PrintedBoardingPass, 64 = AddCustomerId, 65 = DeleteCustomerId, 66 = HoldCreated, 67 = HoldRemoved, 68 = HoldChanged, 69 = OverrideCoupon, 70 = PdsSynchronized, 71 = PdsItemremoved, 72 = Reprice, 73 = ChannelOverride, 74 = EmdCreated, 75 = EmdRemoved, 76 = EmdChanged, 77 = ServiceBundle, 78 = PublishedFareOverride, 79 = FareClassRealignment, 80 = AddedDocument, 81 = ChangedDocument || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/bookings/{bookingKey}/history/bagTagPrint

Summary: Gets the current bag tag print history for a booking, stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
isBagTagSelfPrinted	query	boolean	No	Whether the bag tag is self printed.

Responses

200 Response

404 Response

POST /api/nsk/v1/bookings/{bookingKey}/history/bagTagPrint

Summary: Creates a bag tag print history for a booking, stateless.

Description

Also updates bag's status from Added to AddedPrinted if Self-Tagging Baggage is enabled. In order to see the status of a bag, check bags on a passenger segment.

plaintext



The `Enable Baggage Self-Tagging` in Premium Services and `Write to booking history for bag tag printing` in Departure Control Settings must be enabled in Management Console for self bag-tag printing.

GraphQL endpoints: bookingsBagTagPrintHistoryAdd, bookingsBagTagPrintHistoryAddBatch

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Request Body

The bag tag print history request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Request

Bag Tag Print History Request:

json

```
{
  "legKey": "NjM4NzkzNDMyMDAwMDAwIUI5WITU2MjYhICFTTEMhREVOITMxNDQ0ODk-",
  "receivedBy": "John Doe",
  "passengerName": {
    "first": "John",
    "middle": "Cee",
    "last": "Doe",
    "title": "MR",
    "suffix": null
  },
  "isBagTagSelfPrinted": true,
  "bagTagInformation": [
    {
      "osTag": "0888045757",
      "taggedToStation": "DEN",
      "weight": 15,
      "weightType": 2,
      "originStation": "SLC"
    }
  ]
}
```



Description: Example request for creating a bag tag print history.

Response 400

Null Leg Key:

```
json
{
  "errors": [
    {
      "id": "09ef838d-5e7c-a7fa-fcbf-4b3429d4cfe6",
      "code": "BagTagPrintHistoryRequest.LegKey:RequiredAttribute",
      "message": "BagTagPrintHistoryRequest.LegKey:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "BagTagPrintHistoryRequest",
        "member": "LegKey"
      },
      "rawMessage": "The BagTagPrintHistoryRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The provided leg key is null.

Invalid Leg Key:

```
json
{
  "errors": [
    {
      "id": "cfffb48e-507e-6874-5cbd-a397225ffc9e",
      "code": "nsk:Booking:InvalidLegKey",
      "message": "nsk:Booking:InvalidLegKey",
      "type": "Validation",
      "details": null,
      "rawMessage": "The leg key NjM4NzkzNDMyMDAwMDAwIUSWITU2MjYhICFTTEMhREVOITMxSNDQ0ODk- is not valid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The provided leg key is invalid and cannot be decoded correctly.

Invalid Booking Key:

```
json
{
  "errors": [
    {
      "id": "90465c09-d169-3619-ac68-ed84ca71392e",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'bookingKey' with value 'NDAyMjY2sIVFEUVlGRSFmYWxzZQ--' is invalid.",
      "type": "Validation",
      "details": {
        "bookingKey": "NDAyMjY2sIVFEUVlGRSFmYWxzZQ--"
      },
      "rawMessage": "The identifier 'bookingKey' with value 'NDAyMjY2sIVFEUVlGRSFmYWxzZQ--' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The provided booking key is invalid and cannot be decoded correctly.

Invalid Baggage Weight:

```
json   
  
{  
  "errors": [  
    {  
      "id": "add3f8a9-237d-040c-3a45-e2708bf44a69",  
      "code": "BagTagInformation.Weight:RangeAttribute",  
      "message": "BagTagInformation.Weight:RangeAttribute",  
      "type": "Validation",  
      "details": {  
        "validation": "RangeAttribute",  
        "model": "BagTagInformation",  
        "member": "Weight"  
      },  
      "rawMessage": "The field BagTagInformation must be between 0 and 32767.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The baggage weight is invalid or out of range.

Invalid Baggage Weight Type:

```
json   
  
{  
  "errors": [  
    {  
      "id": "ffedb436-96be-b87c-f338-f68ff1335fcd",  
      "code": "core:Validation:EnumOutOfRange",  
      "message": "The enum 'WeightType' contains an invalid value of '3'.",  
      "type": "Validation",  
      "details": {  
        "value": "3",  
        "name": "WeightType"  
      },  
      "rawMessage": "The enum 'WeightType' contains an invalid value of '3'.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The baggage weight type is invalid or out of range.

Invalid Origin Station Length:

```
json   
  
{  
  "errors": [  
    {  
      "id": "e255e91c-96f7-472f-5662-700ea3f1436d",  
      "code": "BagTagInformation.OriginStation:StringLengthAttribute",  
      "message": "The field BagTagInformation must be a string with a minimum length of 3 and a maximum length of 3.",  
      "type": "Validation",  
      "details": {  
        "validation": "StringLengthAttribute",  
        "model": "BagTagInformation",  
        "member": "OriginStation"  
      },  
      "rawMessage": "The field BagTagInformation must be a string with a minimum length of 3 and a maximum length of 3.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The origin station is exceeded the maximum character length.

Invalid Tagged To Station Length:

```
json

{
  "errors": [
    {
      "id": "6aaa4d83-e780-eb34-028f-b6fbc272d0f9",
      "code": "BagTagInformation.TaggedToStation:StringLengthAttribute",
      "message": "The field BagTagInformation must be a string with a minimum length of 3 and a maximum length of 3.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "BagTagInformation",
        "member": "TaggedToStation"
      },
      "rawMessage": "The field BagTagInformation must be a string with a minimum length of 3 and a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ]
}
```

Description: The tagged to station exceeded the maximum character length.

GET /api/nsk/v1/bookings/{bookingKey}/history/flightMove

Summary: Gets a flight move history stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
LastPageKey	query	string	No	The key of the last booking history item returned.

```
plaintext

|
```

| PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/bookings/{bookingKey}/history/holdDateChange

Summary: Gets the hold date change history records for a booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
Event	query	string	No	The event that triggered the hold date change history entry.

Enumeration values: 66 = Created, 67 = Removed, 68 = Changed || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/bookings/{bookingKey}/history/itinerarySent

Summary: Gets the itinerary sent history records for a booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
LastPageKey	query	string	No	The key of the last booking history item returned.

```
plaintext
|
| PageSize | query | integer | No | The page size for the response. |
```

| PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/bookings/{bookingKey}/history/message

Summary: Gets a booking's message history, stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Responses

200 Response

The list of booking message history.

404 Response

GET /api/nsk/v1/bookings/{bookingKey}/history/notification

Summary: Gets the booking notification history, stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Responses

200 Response

The list of booking notification history.

404 Response

GET /api/nsk/v1/bookings/{bookingKey}/history/seatAssignment

Summary: Gets a seat assignment history stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
Event	query	string	No	The event that triggered the seat assignment history entry.

Enumeration values: 4 = AssignedSeat, 5 = RemoveSeat || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

GET /api/nsk/v1/bookings/{bookingKey}/history/segmentChange

Summary: Gets a segment change history stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.
Event	query	string	No	The event that triggered the seat assignment history entry.

Enumeration values: 6 = AddedFlight, 7 = DeletedFlight || LastPageKey | query | string | No | The key of the last booking history item returned. || PageSize | query | integer | No | The page size for the response. |

Responses

200 Response

Ok.

404 Response

Not found.

POST /api/nsk/v3/bookings/{bookingKey}/queue

Summary: Adds a booking to a booking queue.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Request Body

The booking queue request v2.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 400

Enqueue Booking, Queue Code Does Not Exist:

```
json ✖

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingQueueDoesNotExist",
      "message": "nsk-server:BookingQueueDoesNotExist",
      "type": "Information",
      "details": null,
      "rawMessage": "Booking Queue 'ZZZZZ' does not exist in the DB..",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when trying to add a booking to a queue, and the requested queue code does not exist.

Manual Queueing for Queue Code Not Allowed:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:InvalidSelfServiceRebookingQueueing",
      "message": "nsk-server:InvalidSelfServiceRebookingQueueing",
      "type": "Information",
      "details": null,
      "rawMessage": "Manual queuing from a Self-Service Rebooking Booking Queue is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when trying to add a booking to a queue, and manual queueing from that queue is not allowed.

Booking already in queue:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingAlreadyInQueue",
      "message": "nsk-server:BookingAlreadyInQueue",
      "type": "Information",
      "details": null,
      "rawMessage": "Booking is already on queue: 'DAPI', BookingID = 127160, BookingQueueItemID = 0, SegmentKey = ''.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when trying to add a booking to a queue, and the booking is already in the target queue.

DELETE /api/nsk/v2/bookings/{bookingKey}/queue

Summary: Removes a booking from a booking queue.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Request Body

The booking queue delete request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v1/bookings/{bookingKey}/queue/history

Summary: Gets the booking queue history for a specific booking.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Request Body

The booking queue history request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

404 Response

GET /api/nsk/v1/bookings/{recordLocator}

Summary: Retrieves a booking itinerary stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Responses

200 Response

OK

404 Response

Not found

GET /api/nsk/v1/bookings/{recordLocator}/account

Summary: Retrieves the booking account and collections based on the record locator.

Description

This endpoint allows agents to see all of the different payment account transactions for the booking. See related endpoint for non agents: /booking/payments/bookingCredit

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Responses

404 Response

200 Response

POST /api/nsk/v1/bookings/{recordLocator}/account

Summary: Creates the booking account based on the record locator and data in the request.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Request Body

The create account request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST /api/nsk/v1/bookings/{recordLocator}/account/collection

Summary: Creates a booking account collection and transaction based on the record locator and data in the request.

Description

This endpoint will create a new account collection and a transaction if no collection exists or no matching collection is found. An account collection is unique based on the transaction code and the expiration date. If a matching collection is found, a transaction for that collection is generated and the account collection will be updated. See /api/nsk/v1/resources/accountTransactionCodes for a list of available account transaction codes.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Request Body

The create account collection request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST

[/api/nsk/v1/bookings/{recordLocator}/account/collection/{accountCollectionKey}/transactions](#)

Summary: Creates a booking account collection transaction based on the record locator, the account collection key, and data in the request.

Description

This endpoint will add a transaction to an existing account collection. If there are no account collections, or the account collection key does not match an existing collection, the request will fail. To add a new credit, see [/api/nsk/v1/bookings/{recordLocator}/account/collection](#).

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
accountCollectionKey	path	string	Yes	The account collection key.

Request Body

The create transaction requests.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

GET

[/api/nsk/v2/bookings/{recordLocator}/account/collection/{accountCollectionKey}/transactions](#)

Summary: Retrieves the booking account collection transactions based on the record locator, the account collection key, and the request data.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
accountCollectionKey	path	string	Yes	The account collection key.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

plaintext



|

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json 
```

```
{  
  "data": {  
    "totalCount": 11085,  
    "lastPageKey": "MTAw",  
    "transactions": [  
      {  
        "accountTransactionCode": null,  
        "recordLocator": null,  
        "accountReference": null,  
        "transactionKey": "MSExITEwMjA40CJsJNQxATUzMyEhUURRNUZ",  
        "amount": -2.42,  
        "createdDate": "2025-08-07T04:47:18.2996216Z",  
        "currencyCode": "USD",  
        "foreignAmount": -15.0,  
        "note": "Funds Used: PNR123",  
        "foreignCurrencyCode": "CNY",  
        "type": 0,  
        "paymentId": 365423,  
        "accountCollectionKey": "MSex"  
      }  
    ]  
  }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",  
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "type": "Validation",  
      "details": {  
        "validation": "RequiredAttribute",  
        "model": "TransactionSearchRequestv2",  
        "member": "StartDate"  
      },  
      "rawMessage": "The TransactionSearchRequestv2 field is required.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
  ],  
  "data": null  
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```
json
{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when end date is before start date.

PUT /api/nsk/v1/bookings/{recordLocator}/account/status

Summary: Updates the booking account status.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
status	query	string	No	The allowed account status.
Enumeration values: 0 = Open, 1 = Closed, 2 = AgencyInactive, 3 = Unknown				

Responses

200 Response

500 Response

GET /api/nsk/v2/bookings/{recordLocator}/account/transactions

Summary: Retrieves all of the transactions for the booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

plaintext



|

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

json



```
{  
  "data": {  
    "totalCount": 11085,  
    "lastPageKey": "MTAw",  
    "transactions": [  
      {  
        "accountTransactionCode": null,  
        "recordLocator": null,  
        "accountReference": null,  
        "transactionKey": "MSEXITEwMjA4OCJsjNQxATUzMyEhUURRNUZ",  
        "amount": -2.42,  
        "createdDate": "2025-08-07T04:47:18.2980189Z",  
        "currencyCode": "USD",  
        "foreignAmount": -15.0,  
        "note": "Funds Used: PNR123",  
        "foreignCurrencyCode": "CNY",  
        "type": 0,  
        "paymentId": 365423,  
        "accountCollectionKey": "MSex"  
      }  
    ]  
  }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```

json

{
  "errors": [
    {
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "TransactionSearchRequestv2",
        "member": "StartDate"
      },
      "rawMessage": "The TransactionSearchRequestv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: Example response when start date field is empty.

End Date Before Start Date:

```

json

{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: Example response when end date is before start date.

GET /api/nsk/v1/bookings/{recordLocator}/children

Summary: Gets children record locators of a booking not in state.

Description

A booking gets children bookings as a result of a booking divide operation. This endpoint returns a list of children booking record locators.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	No description

Responses

200 Response

404 Response

POST /api/nsk/v3/bookings/{recordLocator}/comments

Summary: Adds a comment to a booking not in state.

Description

Requires agent permissions. Note: To add comments and log added comments to GDS history please use POST or PUT /api/nsk/v3/booking and set SendToBookingSource to true GDS comments cannot be logged statelessly.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Request Body

The comments to add.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v1/bookings/{recordLocator}/email

Summary: Sends itinerary notification via email.

Description

This method will always send itinerary notifications via email, regardless of the chosen distributionOption on the contact.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Responses

201 Response

HTTP Created status if successful.

POST /api/nsk/v1/bookings/{recordLocator}/fareOverride/journey/{journeyKey}

Summary: Replaces an existing fare price with an amount specified.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking key.
journeyKey	path	string	Yes	The journeyKey.

Request Body

The fare override request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT /api/nsk/v1/bookings/{recordLocator}/fees/{passengerFeeKey}/confirm

Summary: Confirms the passenger fee for a given booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
passengerFeeKey	path	string	Yes	The passenger fee key.

Responses

200 Response

500 Response

PUT /api/nsk/v1/bookings/{recordLocator}/journey

Summary: Moves a journey on a stateless booking to a different journey.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.

Request Body

The move journey request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

POST /api/nsk/v2/bookings/{recordLocator}/notification

Summary: Sends itinerary notification.

Description

Itinerary will be sent using the method set on the contact's distributionOption field.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the booking.

Responses

201 Response

HTTP Created status if successful.

GET /api/dcs/v1/bookings/{recordLocator}/passengers/{passengerKey}/documents/check

Summary: Gets the document check status for a specific passenger if there are any on a specific booking.

Description

Requires a committed booking in state. Requires document check to be enabled in government security configuration. Performs document check and returns the results as documentCheckStatus. More detailed information is intentionally not provided for security reasons. The designers of the client application will need to determine what to do with the different responses, in most cases the passenger will need to be assisted by an airport agent. For more specific requirements check the following endpoint GET/api/nsk/v1/booking/checkin/journey/{journeyKey}/requirements. For configuration options please contact your CSDM. For full documentation see <https://developer.navitaire.com/documentation/dotrez-api/how-to/adc>.

plaintext 

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the associated booking.
passengerKey	path	string	Yes	The passenger key.

Responses

200 Response

404 Response

GET /api/dcs/v1/bookings/{recordLocator}/passengers/{passengerKey}/infant/documents/check

Summary: Gets the document check status for a specific passenger's infant if there are any on a specific booking.

Description

Requires a committed booking in state. Requires document check to be enabled in government security configuration. Performs document check and returns the results as documentCheckStatus. More detailed information is intentionally not provided for security reasons. The designers of the client application will need to determine what to do with the different responses, in most cases the passenger will need to be assisted by an airport agent. For more specific requirements check the following endpoint GET/api/nsk/v1/booking/checkin/journey/{journeyKey}/requirements. For configuration options please contact your CSDM. For full documentation see <https://developer.navitaire.com/documentation/dotrez-api/how-to/adc>.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the associated booking.
passengerKey	path	string	Yes	The passenger key.

Responses

200 Response

404 Response

POST /api/nsk/v1/bookings/{recordLocator}/passengers/{passengerKey}/seats/{unitKey}

Summary: Agent only. Adds a seat assignment to a specific passenger on a booking.

Description

Agent only. This endpoint actively checks whether the passenger has a seat. If the passenger does have a seat, it will delete the current seat the passenger occupies and add the requested one. This may result in slightly slower response times.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

GraphQL endpoints: bookingsSeatAdd, bookingsSeatsAdd

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
passengerKey	path	string	Yes	The passenger key.
unitKey	path	string	Yes	The unit key.

Request Body

The add seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

DELETE /api/nsk/v1/bookings/{recordLocator}/passengers/{passengerKey}/seats/{unitKey}

Summary: Agent only. Deletes a seat assignment from a specific passenger on a booking.

Description

Agent only.

plaintext	
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.	

GraphQL endpoints: bookingsSeatDelete, bookingsSeatsDelete

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
passengerKey	path	string	Yes	The passenger key.
unitKey	path	string	Yes	The unique unit key.

Request Body

The delete seat request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v1/bookings/{recordLocator}/queues/{queueCode}

Summary: Retrieves the details for the booking queues of a specific booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator.
queueCode	path	string	Yes	The queue code.

Request Body

The booking queue request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

DELETE /api/nsk/v1/bookings/checkin/{recordLocator}/journey/{journeyKey}

Summary: Removes the checkin status of passengers for a specific journey stateless.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking record locator to look up the journey.
journeyKey	path	string	Yes	The journey key.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v3/bookings/checkin/{recordLocator}/journey/{journeyKey}

Summary: Checks in a collection of passengers to a specific journey stateless.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

- When debug mode is disabled, warning and error messages will be masked if the `SystemOptions.HideErrorMessages` is set to true.
- When debug mode is disabled, warning and error messages will be returned to agents only if the `SystemOptions.HideErrorMessages` is set to false.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking record locator to look up the journey.
journeyKey	path	string	Yes	The journey key of the journey to check into.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Ok if checkin was successful.

400 Response

Examples

Response 201

Check-in Successful:

```
json ✖  
{  
  "data": null  
}
```

Description: Example when check-in is successful.

Checkin Valid with displaying warning message to agents.: ✖

```
json
{
  "messages": [
    {
      "code": "CheckinPassengerWarning",
      "type": "Warning",
      "value": "AUT TEST Authorization Response Free Text\nn8501:AU:OK TO BOARD\nnDocuments are required for this flight.",
      "status": 0,
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_MDgvMjEvMjAyNCAwOTowMH5TWUR_MDgvMjEvMjAyNCAxMjowMH5_"
      }
    },
    "data": null
  }
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This requires Digital API configuration to be set such that warning messages can be returned to agents only when debug is turned off.

Checkin Valid with a masked warning message::

```
json
{
  "messages": [
    {
      "code": "CheckinPassengerWarning",
      "type": "Warning",
      "value": "CheckinPassengerWarning",
      "status": 0,
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_MDgvMjEvMjAyNCAwOTowMH5TWUR_MDgvMjEvMjAyNCAxMjowMH5_"
      }
    },
    "data": null
  }
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This is what non-agents will see regardless of Digital API configuration for displaying warning messages when debug is off. This is also what an agent would see if the corresponding API configuration is not set to allow warning messages to be shown.

Response 400

Checkin Invalid:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:Invalid",
      "message": "nsk:Checkin:Invalid",
      "type": "Validation",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Checkin request is invalid. Please refer to checkin requirements for more details.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  }
}
```

Description: Example error when check in requirements are not met. The /requirements endpoint should be called before attempting a checkin.

Seat Missing:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1223",
      "message": "nsk:Checkin:TypeID:1223",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Baggins/Bilbo missing seat assignments on flight number 2452.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: In this example the optional request parameter `seatsRequired` was set to true. One passenger had a seat assigned and one did not.

Inhibited:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3096",
      "message": "nsk:Checkin:TypeID:3096",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3088",
      "message": "nsk:Checkin:TypeID:3088",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: In this example the passenger meets all the requirements but is inhibited due to government security.

Missing verified document with agent visible error message:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1236",
      "message": "Baggins/Bilbo missing verified documents.",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: This example shows both the expected response when a passenger is missing verified documents, as well as what a logged-in agent would see if the API configuration is set correctly. This is an example response when debug mode is turned off.

GET /api/nsk/v2/bookings/checkin/{recordLocator}/journey/{journeyKey}/requirements

Summary: Gets the checkin pre-validation requirements for a specific journey version 2.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the booking.
journeyKey	path	string	Yes	The journey key to be pre validate checkin with.

Responses

200 Response

The checkin validation requirements.

404 Response

409 Response

Examples

Response 200

Valid Response No Restrictions:

json



```
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": {
          "documentsOnFile": [
            "P",
            "V"
          ],
          "governmentDocumentInstances": [
            {
              "governmentInstance": "USDHS",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                },
                {
                  "group": 5,
                  "documentTypeCode": "M"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            },
            {
              "governmentInstance": "CBIA",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            }
          ],
          "isValid": true
        },
        "addressRequirements": {
          "isValid": false,
          "addressesOnFile": [
            "Address1"
          ],
          "unhandledAddresses": [
            "Address2"
          ],
          "invalidSsrs": null,
          "requiredSsrs": null,
        }
      }
    }
  }
}
```

```

        "restrictions": null,
        "isValid": true
    },
    "governmentProgramRequirements": {
        "usdhs": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "USDHS",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        },
        "cbsa": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "CBSA",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        }
    },
    "restrictions": null,
    "isValid": true
}
}

```

Description: Example Response

Response with Restrictions:

```

json
{
    "data": {
        "passengers": {
            "passengerKey": {
                "passengerKey": "PassengerKey",
                "documentRequirements": null,
                "addressRequirements": null,
                "invalidSsrs": null,
                "requiredSsrs": null,
                "restrictions": [
                    {
                        "restriction": 12,
                        "description": "Missing Documents"
                    },
                    {
                        "restriction": 1,
                        "description": "Missing Destination Address"
                    }
                ],
                "isValid": false
            }
        },
        "governmentProgramRequirements": {
            "usdhs": {
                "governmentInstance": null,
                "requiredDocumentFields": null,
                "additionalDataRequired": null,
                "programDescription": "USDHS",
                "messageType": null,
                "minVerifiedDocumentCount": 1
            }
        },
        "restrictions": [
            4
        ],
        "isValid": false
    }
}

```

Description: Example Response

GET /api/nsk/v1/bookings/checkin/{recordLocator}/journey/{journeyKey}/status

Summary: Gets the passengers lift status stateless for a specific journey based on the record locator.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator in reference to the journey.
journeyKey	path	string	Yes	The journey key in reference.

Responses

200 Response

The collection of passenger lift statuses for the journey.

404 Response

DELETE /api/nsk/v1/bookings/checkin/{recordLocator}/segment/{segmentKey}

Summary: Removes the checkin status of passengers for a specific journeys segment stateless.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking record locator to look up the journeys segment.
segmentKey	path	string	Yes	The segment key.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

POST /api/nsk/v3/bookings/checkin/{recordLocator}/segment/{segmentKey}

Summary: Checks in a collection of passengers to a specific journeys segment stateless.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

- When debug mode is disabled, warning and error messages will be masked if the `SystemOptions.HideErrorMessages` is set to true.
- When debug mode is disabled, warning and error messages will be returned to agents only if the `SystemOptions.HideErrorMessages` is set to false.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The booking record locator to look up the journeys segment.
segmentKey	path	string	Yes	The segment key of the segment to check into.

Request Body

The checkin passengers request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Ok if checkin was successful.

400 Response

Examples

Response 201

Check-in Successful:

```
json   
{  
  "data": null  
}
```

Description: Example when check-in is successful.

Checkin Valid with displaying warning message to agents.:.

```
json   
{  
  "messages": [  
    {  
      "code": "CheckinPassengerWarning",  
      "type": "Warning",  
      "value": "AUT TEST Authorization Response Free Text\n8501:AU:OK TO BOARD\nDocuments are required for this flight.",  
      "status": 0,  
      "details": {  
        "passengerKey": "MCFBRFQ-",  
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_mDgvMjEvMjAyNCAwOTowMH5TWUR_mDgvMjEvMjAyNCAxMjowMH5_"  
      }  
    },  
    "data": null  
  ]  
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This requires Digital API configuration to be set such that warning messages can be returned to agents only when debug is turned off.

Checkin Valid with a masked warning message.:.

```
json
```

```
{  
  "messages": [  
    {  
      "code": "CheckinPassengerWarning",  
      "type": "Warning",  
      "value": "CheckinPassengerWarning",  
      "status": 0,  
      "details": {  
        "passengerKey": "MCFBRFQ-",  
        "segmentKey": "T1Z_NTYzMH4gfn5CTkV_mDgvMjEvMjAyNCAwOTowMH5TWUR_mDgvMjEvMjAyNCAxMjowMH5_"  
      }  
    },  
    "data": null  
  ]  
}
```

Description: Example message when check in requirements are met, but there are still travel documents required. This is what non-agents will see regardless of Digital API configuration for displaying warning messages when debug is off. This is also what an agent would see if the corresponding API configuration is not set to allow warning messages to be shown.

Response 400

Checkin Invalid:

```
json
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Checkin:Invalid",  
      "message": "nsk:Checkin:Invalid",  
      "type": "Validation",  
      "details": {  
        "passengerKey": "MCFBRFQ-",  
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_mDkvMDYvMjAyNCAwOTowMH5NTkx_mDkvMDcvMjAyNCAwNTowMH5_"  
      },  
      "rawMessage": "Checkin request is invalid. Please refer to checkin requirements for more details.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
    "data": null  
  ]  
}
```

Description: Example error when check in requirements are not met. The /requirements endpoint should be called before attempting a checkin.

Seat Missing:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1223",
      "message": "nsk:Checkin:TypeID:1223",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "Baggins/Bilbo missing seat assignments on flight number 2452.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "data": null
    }
  ]
}
```

Description: In this example the optional request parameter `seatsRequired` was set to true. One passenger had a seat assigned and one did not.

Inhibited:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3096",
      "message": "nsk:Checkin:TypeID:3096",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:3088",
      "message": "nsk:Checkin:TypeID:3088",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwNTowMH5_"
      },
      "rawMessage": "United States - SecureFlight: Passenger inhibited from checkin. CONTACT THE DHS RESOLUTION DESK AT: 1-800-CALL-1-800",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    },
    {
      "data": null
    }
  ]
}
```

Description: In this example the passenger meets all the requirements but is inhibited due to government security.

Missing verified document with agent visible error message:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Checkin:TypeID:1236",
      "message": "Baggins/Bilbo missing verified documents.",
      "type": "Error",
      "details": {
        "passengerKey": "MCFBRFQ-",
        "segmentKey": "T1Z_NTY1NX4gfn5TTEN_MDkvMDYvMjAyNCAwOTowMH5NTkx_MDkvMDcvMjAyNCAwOTowMH5_"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: This example shows both the expected response when a passenger is missing verified documents, as well as what a logged-in agent would see if the API configuration is set correctly. This is an example response when debug mode is turned off.

GET /api/nsk/v2/bookings/checkin/{recordLocator}/segment/{segmentKey}/requirements

Summary: Gets the checkin pre-validation requirements for a specific bookings journeys segment version 2.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the booking.
segmentKey	path	string	Yes	The segment key to pre validate checkin with.

Responses

200 Response

The checkin validation requirements.

404 Response

409 Response

Examples

Response 200

Valid Response No Restrictions:

json



```
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": {
          "documentsOnFile": [
            "P",
            "V"
          ],
          "governmentDocumentInstances": [
            {
              "governmentInstance": "USDHS",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                },
                {
                  "group": 5,
                  "documentTypeCode": "M"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            },
            {
              "governmentInstance": "CBIA",
              "numberOfMissingDocuments": 0,
              "eligibleDocuments": [
                {
                  "group": 18,
                  "documentTypeCode": "I"
                }
              ],
              "verifiableDocuments": [
                {
                  "group": 6,
                  "documentTypeCode": "P"
                },
                {
                  "group": 9,
                  "documentTypeCode": "V"
                }
              ],
              "documentVerificationRequired": true
            }
          ],
          "isValid": true
        },
        "addressRequirements": {
          "isValid": false,
          "addressesOnFile": [
            "Address1"
          ],
          "unhandledAddresses": [
            "Address2"
          ],
          "invalidSsrs": null,
          "requiredSsrs": null,
        }
      }
    }
  }
}
```

```

        "restrictions": null,
        "isValid": true
    },
    "governmentProgramRequirements": {
        "usdhs": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "USDHS",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        },
        "cbsa": {
            "governmentInstance": null,
            "requiredDocumentFields": null,
            "additionalDataRequired": null,
            "programDescription": "CBSA",
            "messageType": null,
            "minVerifiedDocumentCount": 1
        }
    },
    "restrictions": null,
    "isValid": true
}
}

```

Description: Example Response

Response with Restrictions:

```

json
{
  "data": {
    "passengers": {
      "passengerKey": {
        "passengerKey": "PassengerKey",
        "documentRequirements": null,
        "addressRequirements": null,
        "invalidSsrs": null,
        "requiredSsrs": null,
        "restrictions": [
          {
            "restriction": 12,
            "description": "Missing Documents"
          },
          {
            "restriction": 1,
            "description": "Missing Destination Address"
          }
        ],
        "isValid": false
      }
    },
    "governmentProgramRequirements": {
      "usdhs": {
        "governmentInstance": null,
        "requiredDocumentFields": null,
        "additionalDataRequired": null,
        "programDescription": "USDHS",
        "messageType": null,
        "minVerifiedDocumentCount": 1
      }
    },
    "restrictions": [
      4
    ],
    "isValid": false
  }
}

```

Description: Example Response

GET /api/nsk/v1/bookings/checkin/{recordLocator}/segment/{segmentKey}/status

Summary: Gets the passengers lift status stateless for a specific segment based on the record locator.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator in reference to the journey.
segmentKey	path	string	Yes	The segment key in reference.

Responses

200 Response

The collection of passenger lift statuses for the journey.

404 Response

POST /api/nsk/v1/bookings/queues

Summary: Queues the bookings by inventory leg.

Description

Request Details

Request Body

The queue booking by inventory leg request

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 201

Queued bookings successfully:

```
json
{
  "data": null
}
```

Description: Example when all the bookings are queued successfully.

Queued with skipped bookings:

```
json
```

```
{  
    "messages": [  
        {  
            "code": "nsk:BookingQueues:BookingsNotQueued",  
            "type": "Warning",  
            "value": "There are bookings that were not queued.",  
            "status": 1,  
            "details": {  
                "skippedBookings": "ZB3LMA,Y141SB,0232XL"  
            }  
        },  
    ],  
    "data": null  
}
```

Description: Example when there are bookings skipped during inventory queueing due to unhandled exceptions.

Response 400

Leg Key is null or empty:

```
json
```

```
{  
    "errors": [  
        {  
            "id": "bef37773-8bc1-0d5e-b407-287277dfee16",  
            "code": "QueueBookingByInventoryLegRequest.LegKey:RequiredAttribute",  
            "message": "QueueBookingByInventoryLegRequest.LegKey:RequiredAttribute",  
            "type": "Validation",  
            "details": {  
                "validation": "RequiredAttribute",  
                "model": "QueueBookingByInventoryLegRequest",  
                "member": "LegKey"  
            },  
            "rawMessage": "The QueueBookingByInventoryLegRequest field is required.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        },  
    ],  
    "data": null  
}
```

Description: Example when the Leg Key is null or empty.

Booking Queue Codes is null or empty:

```
json
```

```
{  
    "errors": [  
        {  
            "id": "b5b77041-86b5-d75b-5685-03d7d09c7591",  
            "code": "QueueBookingByInventoryLegRequest.BookingQueueCodes:RequiredAttribute",  
            "message": "QueueBookingByInventoryLegRequest.BookingQueueCodes:RequiredAttribute",  
            "type": "Validation",  
            "details": {  
                "validation": "RequiredAttribute",  
                "model": "QueueBookingByInventoryLegRequest",  
                "member": "BookingQueueCodes"  
            },  
            "rawMessage": "The BookingQueueCodes field is required. It is empty or only contains null values.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        },  
    ],  
    "data": null  
}
```

Description: Example when the Booking Queue Codes is null or empty.

At least 1 null or empty booking queue code:

```
json

{
  "errors": [
    {
      "id": "ff8a93b6-79d0-051e-67eb-1e69fbde1aa9",
      "code": "nsk:BookingQueues:BookingQueueCodesNullOrEmpty",
      "message": "nsk:BookingQueues:BookingQueueCodesNullOrEmpty",
      "type": "Validation",
      "details": null,
      "rawMessage": "Booking queue codes cannot be null or empty.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if at least one of the Booking Queue Codes is null or empty.

Invalid Leg Key:

```
json

{
  "errors": [
    {
      "id": "399ce6c5-16f7-9a6a-c234-49bb51e0070c",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "legKey": "abc"
      },
      "rawMessage": "The identifier 'legKey' with value 'abc' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if the Leg Key is not valid.

Comments is null or empty:

```
json

{
  "errors": [
    {
      "id": "0a4856dc-2a3b-b176-b276-5d89b2dbc1da",
      "code": "QueueBookingByInventoryLegRequest.Comments:RequiredAttribute",
      "message": "QueueBookingByInventoryLegRequest.Comments:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "QueueBookingByInventoryLegRequest",
        "member": "Comments"
      },
      "rawMessage": "The QueueBookingByInventoryLegRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the comments is null or empty.

Inventory Queuing is not allowed:

```
json

{
  "errors": [
    {
      "id": "35c280c4-26e4-3357-e9f3-c05fd9c3d412",
      "code": "nsk:BookingQueues:QueueBookingsOnSpecifiedLegNotAllowed",
      "message": "nsk:BookingQueues:QueueBookingsOnSpecifiedLegNotAllowed",
      "type": "Error",
      "details": null,
      "rawMessage": "Queuing bookings on the specified leg is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if queuing bookings on the specified leg is not allowed.

Flight is before the allowed time:

```
json

{
  "errors": [
    {
      "id": "a01958be-bd6c-bdad-ec31-af616b1a0c8b",
      "code": "nsk-server:InventoryQueuingNotAllowed",
      "message": "nsk-server:InventoryQueuingNotAllowed",
      "type": "Error",
      "details": {
        "message": "Selected flight is no longer valid for flight queuing."
      },
      "rawMessage": "Selected flight is no longer valid for flight queuing.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if the flight is before the allowed time to process.

Flight is after the allowed time:

```
json

{
  "errors": [
    {
      "id": "a01958be-bd6c-bdad-ec31-af616b1a0c8b",
      "code": "nsk-server:InventoryQueuingNotAllowed",
      "message": "nsk-server:InventoryQueuingNotAllowed",
      "type": "Error",
      "details": {
        "message": "Selected flight is not yet valid for flight queuing."
      },
      "rawMessage": "Selected flight is not yet valid for flight queuing.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if the flight is after the allowed time to process.

Flight does not contain any bookings:

```
json

{
  "errors": [
    {
      "id": "741c2461-9eaa-31f8-2868-d43a3aee913e",
      "code": "nsk-server:InventoryQueuingNotAllowed",
      "message": "nsk-server:InventoryQueuingNotAllowed",
      "type": "Error",
      "details": {
        "message": "Selected flight does not contain any bookings."
      },
      "rawMessage": "Selected flight does not contain any bookings.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if the flight does not contain any bookings

Booking queue code does not exist:

```
json

{
  "errors": [
    {
      "id": "8b150b05-cbbb-2623-c612-31d6b4650e3f",
      "code": "nsk-server:InventoryQueuingNotAllowed",
      "message": "nsk-server:InventoryQueuingNotAllowed",
      "type": "Error",
      "details": {
        "message": "Selected booking queue DAPI is not allowed for flight queuing."
      },
      "rawMessage": "Selected booking queue DAPI is not allowed for flight queuing.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if a booking queue code in the list does not exist.

Booking queue code is not of Inventory Queuing Event Type:

```
json

{
  "errors": [
    {
      "id": "9e497521-d2e2-1f2d-3bb8-caa77787db1a",
      "code": "nsk-server:InventoryQueuingNotAllowed",
      "message": "nsk-server:InventoryQueuingNotAllowed",
      "type": "Error",
      "details": {
        "message": "Selected booking queue AQCD is not allowed for flight queuing."
      },
      "rawMessage": "Selected booking queue AQCD is not allowed for flight queuing.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example if a booking queue code in the list is not of Inventory Queuing Event Type.

Flight does not exist:

```
json   
{  
  "errors": [  
    {  
      "id": "cef3c8a5-a9d5-21b1-493e-e3fedd7cc18f",  
      "code": "nsk-server:InventoryQueuingNotAllowed",  
      "message": "nsk-server:InventoryQueuingNotAllowed",  
      "type": "Error",  
      "details": {  
        "message": "Flight does not exist."  
      },  
      "rawMessage": "Flight does not exist.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example if the provided Leg Key has an nonexistent flight.

DELETE /api/nsk/v1/bookings/queues/{bookingQueueKey}

Summary: Removes a booking from the specified booking queue item.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueKey	path	string	Yes	The unique key of the booking queue to dequeue from.

Request Body

The request for the dequeue operation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/nsk/v2/bookings/queues/{queueCode}/items

Summary: Retrieves the booking queue items associated with the queue code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
queueCode	path	string	Yes	The queue code.
SubQueueCode	query	string	No	The sub queue code.

```
plaintext   
|
```

| StartDate | query | string (date-time) | No | The start date. || EndDate | query | string (date-time) | No | The end date. || QueueEventType | query | string | No | The queue event type.

Enumeration values: 0 = Default, 1 = BookingBalanceDue, 2 = BookingNegativeBalance, 3 = BookingCustomerComment, 4 = DeclinedPaymentInitial, 5 = DeclinedPaymentChange, 6 = FareOverride, 7 = ScheduleTimeChange, 8 = ScheduleTimeChangeMisconnect, 9 = ScheduleCancellation, 10 = FlightDesignatorChange, 11 = ReaccommodationMove, 12 = GdsCancelWithPendingPayment, 13 = InvalidPriceStatusOverride, 14 = FareRestrictionOverride, 15 = HeldBookings, 16 = InvalidPriceStatus, 17 = Watchlist, 18 = NonFlightServiceFee, 19 = NotAllTicketNumbersReceived, 20 = BookingSegmentOversold, 21 = ReaccommodationCancel, 22 = ExternalSsrAutoConfirmed, 23 = OpCarrierSegUpdate, 24 = OpCarrierSsrUpdate, 25 = OpCarrierOtherUpdate, 26 = NameChangeNotAllowed, 27 = InboundAscNotProcessed, 28 = OpCarrierInformationChange, 29 = BookingComponentUpdate, 30 = GroupBookings, 31 = BankDirectPnrOutOfBalance, 32 = NoSeatAssigned, 33 = SeatNumberChange, 34 = SsrNotSupportedOnNewSeat, 35 = FewerSeatPreferencesMetOnNewSeat, 36 = AosUnableToConfirmCancel, 37 = ETicketIssue, 38 = ETicketFollowup, 39 = InvoluntaryFlyAhead, 40 = ManualClearanceOnOutage, 41 = UnbalancedPoints, 42 = OpCarrierTimeChange, 43 = OaCarrierTimeChange, 44 = VoluntaryFlightChange, 45 = InvoluntaryFlightChange, 46 = HoldCancellationFailed, 47 = ScheduleTimeChangeWithDynamicQueueCode, 48 = ReaccommodationMoveWithDynamicQueueCode, 49 = ItineraryIntegrity, 50 = ReducePartyNotProcessed, 51 = CheckedPassengerUpdate, 52 = MustBeSeatGroupViolation, 57 = NameChangeWithinRule, 58 = IncompletePassengerEMDCoupon, 59 = ASVCUpdateFailed, 60 = FailedFareClassRealignment, 61 = ScheduleTimeChangeMisconnectBelowMinimum, 62 = ScheduleTimeChangeMisconnectAboveMaximum, 63 = OpCarrierMisconnect, 64 = InventoryQueuing, 65 = SelfServiceAsm, 66 = SelfServiceDelay, 67 = SelfServiceFlightCancellation, 68 = SelfServiceDrop, 69 = SelfServiceScheduleChange, 70 = SelfServiceTimeChange, 71 = AosAddOrCancelNotSuccessful, 72 = MissingGdsFareFamily, 73 = ChangeCabinQueue, 74 = PendingPaymentQueue, 75 = TicketDisplayToDetermineIncludedServiceBundleFailed, 76 = DuplicateTicketsInBooking, 77 = TicketCouponUpdateQueue, 78 = ETicketRefund || PageSize | query | integer | No | The number of records to be retrieved per page. To retrieve the next set of records, use the endpoint again with the lastIndex set. || LastIndex | query | integer (int64) | No | The last row retrieved. This is used for paging. |

Responses

200 Response

404 Response

POST /api/nsk/v2/bookings/quote

Summary: Retrieves a itinerary quote based on the request.

Description

Although this endpoint behaves like a GET, POST is used to reduce the limitations resulting from URL length issues. Itinerary is a stateless booking and not tied to the logged in user.

Request Details

Request Body

The itinerary quote request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Itinerary returned OK

400 Response

There was a problem with the request and it could not be completed.

404 Response

The item was not found.

GET /api/nsk/v2/bookings/searchByAgency

Summary: Find booking by agency.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
OrganizationCode	query	string	No	OrganizationCode represents the organization associated with the agent who created the booking.

```
plaintext 
```

| FirstName | query | string | No | The first name of a passenger. || LastName | query | string | No | The last name of a passenger. || PhoneticSearch | query | boolean | No | Whether or not to do a phonetic search of the last name. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByAgent

Summary: Find booking by agent.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
AgentId	query	integer (int64)	No	The agent identifier.

```
plaintext 
```

| FirstName | query | string | No | The first name of a passenger. || LastName | query | string | No | The last name of a passenger. || PhoneticSearch | query | boolean | No | Whether or not to do a phonetic search. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByAgentCode

Summary: Find booking by the agent code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
AgentCode	query	string	No	The agent code (name) that created the booking.

```
plaintext |
```

| DomainCode | query | string | No | The domain code to find. || FirstName | query | string | No | The first name of a passenger. || LastName | query | string | No | The last name of a passenger. || PhoneticSearch | query | boolean | No | Whether or not to do a phonetic search. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v3/bookings/searchByBagTag

Summary: Find booking by a bag tag. The StartUtc needs to be within 7 days before the created date of the booking. If the StartUtc is set to DateTime.MinValue (1/1/0001) or not set at all, the value will be set to the current date, UTC.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
BagTag	query	string	No	One of the OS tag number.

```
plaintext |
```

| StartUtc | query | string (date-time) | No | The OS tag start date. If not set or set to DateTime.MinValue (1/1/0001), will default to current date, UTC. This date needs to be within 7 days before the created date of the booking. || PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). || SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

The booking response.

404 Response

GET /api/nsk/v2/bookings/searchByContact

Summary: Find booking by contact.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
FirstName	query	string	No	The first name of a passenger or contact.

```
plaintext |
```

| LastName | query | string | No | The last name of a passenger or contact. || RecordLocator | query | string | No | Record locator being requested. || PhoneNumber | query | string | No | One of the applied phone numbers to a passenger or contact. || EmailAddress | query | string | No | The email address of a contact. || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationCode | query | string | No | OrganizationCode represents the organization associated with the agent who created the booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || SearchArchive | query | boolean | No | Whether or not to search the booking archive. || PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByContactCustomerNumber

Summary: Find booking by contact customer number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ContactCustomerNumber	query	string	No	One of the contacts customer number applied.

```
plaintext |
```

| AgentId | query | integer (int64) | No | The agent identifier. || OrganizationCode | query | string | No | OrganizationCode represents the organization associated with the agent who created the booking. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

POST /api/nsk/v1/bookings/searchByCreditCard

Summary: Find booking by credit card number.

Description

This endpoint behaves like a GET but is masked as a POST.

Request Details

Request Body

Search bookings by credit card request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v2/bookings/searchByCustomerNumber

Summary: Find booking by customer number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CustomerNumber	query	string	No	One of the contacts customer number applied.

plaintext



|

| AgentId | query | integer (int64) | No | The agent identifier. || OrganizationCode | query | string | No | OrganizationCode represents the organization associated with the agent who created the booking. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByDate

Summary: Find booking by date.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
StartDateUtc	query	string (date-time)	No	The booking date start search.

```
plaintext |
```

| EndDateUtc | query | string (date-time) | No | The booking date end search. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByDocument

Summary: Find booking by document.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Document.DocumentNumber	query	string	No	One of the passengers applied travel document number.

```
plaintext |
```

| Document.DocumentTypeCode | query | string | No | One of the passengers applied travel document type code. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByEmail

Summary: Find booking by email.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
EmailAddress	query	string (email)	No	The email address of a contact.

```
plaintext 
```

| AgentId | query | integer (int64) | No | The agent identifier. || PhoneticSearch | query | boolean | No | Whether or not to do a phonetic search. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v1/bookings/searchByExternalPayment

Summary: Used to search the list of bookings that have used the account from the payment. Only External Payments should be used for searching.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
RecordLocator	query	string	No	The record locator of a booking.

```
plaintext 
```

| PaymentKey | query | string | No | The payment key of a specific payment under a booking. || PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). || FlightNumber | query | string | No | Filter by the flight number. || DepartureDate | query | string (date-time) | No | The first journeys departure date. || Destination | query | string | No | Filter by the destination code. || Origin | query | string | No | Filter by the departure code. || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. |

Responses

200 Response

404 Response

GET /api/nsk/v2/bookings/searchByLastName

Summary: Find booking by last name.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
LastName	query	string	No	The last name of a passenger or contact.

```
plaintext |
```

| FirstName | query | string | No | The first name of a passenger or contact. || PhoneticSearch | query | boolean | No | Whether or not to do a phonetic search. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByPhone

Summary: Find booking by phone.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
PhoneNumber	query	string	No	One of the applied phone numbers to a passenger or contact.

```
plaintext |
```

| AgentId | query | integer (int64) | No | The agent identifier. || OrganizationCode | query | string | No | OrganizationCode represents the organization associated with the agent who created the booking. || Filters.PageSize | query | integer | No | The number of items to return for the request. || Filters.LastIndex | query | integer (int64) | No | The last booking index (used for paging). || Filters.FlightNumber | query | string | No | Filter by the flight number. || Filters.DepartureDate | query | string (date-time) | No | The first journeys departure date. || Filters.Destination | query | string | No | Filter by the destination code. || Filters.Origin | query | string | No | Filter by the departure code. || Filters.SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || Filters.OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || Filters.SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByRecordLocator

Summary: Find booking by record locator.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
RecordLocator	query	string	No	Record locator being requested.

```
plaintext 
```

| PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

OK

404 Response

Not found.

GET /api/nsk/v2/bookings/searchByReferenceNumber

Summary: Find booking by reference number. The reference number is used as identifier by New Skies Rail customers for purposes of ticketing and check in.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ReferenceNumber	query	integer (int64)	No	The reference number, used as an identifier by New Skies Rail customers for purposes of ticketing and check in.

```
plaintext 
```

| AgentId | query | integer (int64) | No | The agent identifier. || OrganizationCode | query | string | No | The organization code associated with the agent who created the booking. || PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

The booking response.

404 Response

GET /api/nsk/v3/bookings/searchByThirdPartyRecordLocator

Summary: Find booking by a third party record locator.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
SystemCode	query	string	No	The system code to find.

plaintext 

| AgentId | query | integer (int64) | No | The agent identifier. || OrganizationCode | query | string | No | OrganizationCode represents the organization associated with the agent who created the booking. || RecordLocator | query | string | No | Record locator being requested. || PageSize | query | integer | No | The number of items to return for the request. || LastIndex | query | integer (int64) | No | The last booking index (used for paging). || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. |

Responses

200 Response

The booking response.

404 Response

GET /api/nsk/v1/bookings/watchList/{bookingKey}

Summary: Retrieves the watch list for a specific booking.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
bookingKey	path	string	Yes	The booking key.

Responses

200 Response

404 Response

Bundles Endpoints

GET /api/nsk/v1/bundles/applications/{bundleApplicationKey}/details

Summary: Retrieves a specific bundle application's details.

Description

See `api/nsk/v1/resources/bundles/applications` to retrieve the collection of bundle application resources.

Request Details

Parameters

Name	Location	Type	Required	Description
bundleApplicationKey	path	string	Yes	The bundle application key.

Responses

404 Response

200 Response

GET /api/nsk/v1/bundles/rules/{bundleRuleCode}/details

Summary: Retrieves the list of bundle applications associated with the bundle rule.

Description

See `api/nsk/v1/resources/bundles/rules` to retrieve the collection of bundle rule resources.

Request Details

Parameters

Name	Location	Type	Required	Description
bundleRuleCode	path	string	Yes	The bundle rule code.

Responses

404 Response

200 Response

GET /api/nsk/v1/bundles/sets/{bundleSetCode}/details

Summary: Retrieves the list of bundle codes associated with the bundle set.

Description

See `api/nsk/v1/resources/bundles/sets` to retrieve the collection of bundle set resources.

Request Details

Parameters

Name	Location	Type	Required	Description
bundleSetCode	path	string	Yes	The bundle set code.

Responses

404 Response

200 Response

Collection Endpoints

PUT /api/nsk/v1/collection/{accountCollectionKey}/expirationDate

Summary: Updates the expiration of the account collection based on the account collection key and data in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
accountCollectionKey	path	string	Yes	The account collection key.

Request Body

The account expiration update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

GET /api/nsk/v2/collection/{accountCollectionKey}/transactions

Summary: Retrieves the credit transactions based on the account collection key and the request data.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
accountCollectionKey	path	string	Yes	The account collection key.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

plaintext



|

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json 
```

```
{  
    "data": {  
        "totalCount": 11085,  
        "lastPageKey": "MTAw",  
        "transactions": [  
            {  
                "accountTransactionCode": null,  
                "recordLocator": null,  
                "accountReference": null,  
                "transactionKey": "MSExITEwMjA4OCJsJNQxATUzMyEhUURRNUZ",  
                "amount": -2.42,  
                "createdDate": "2025-08-07T04:47:18.2941617Z",  
                "currencyCode": "USD",  
                "foreignAmount": -15.0,  
                "note": "Funds Used: PNR123",  
                "foreignCurrencyCode": "CNY",  
                "type": 0,  
                "paymentId": 365423,  
                "accountCollectionKey": "MSex"  
            }  
        ]  
    }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json 
```

```
{  
    "errors": [  
        {  
            "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",  
            "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
            "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
            "type": "Validation",  
            "details": {  
                "validation": "RequiredAttribute",  
                "model": "TransactionSearchRequestv2",  
                "member": "StartDate"  
            },  
            "rawMessage": "The TransactionSearchRequestv2 field is required.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        }  
    ],  
    "data": null  
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```

json

{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  }
}

```

Description: Example response when end date is before start date.

Contacts Endpoints

GET /api/nsk/v1/contacts/searchByPhoneNumber

Summary: Gets the latest contact in a booking with a specific phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
PhoneNumber	query	string	No	The phone number.

```

plaintext
|

```

Responses

200 Response

404 Response

Contents Endpoints

GET /api/nsk/v1/contents/news

Summary: Retrieves the available news metadata.

Description

News or sometimes referred to as FYI Content notes are created by management about your carrier operations. They can be viewed by reservation agents and station agents.

News notes are a quick way for airlines to distribute important information to reservation and check-in agents.

If your carrier has a new fare rule, you can use FYI Content to enter this information and set the FYI type to specify how and when this information is displayed. You can also determine who reads this information by assigning roles to it.

You can also use news content to store pertinent station info, such as passenger disruption reports or journal entries. By saving the content as type No Alert, you leave it in the database, where other users can access it with a search. You can group station reports by Domain, Department, Role, Date, Name, and so on.

Related endpoints:

- `GET /resources/contents` which contains generic data for all content types including news contents
- `GET /resources/contents/{contentID}/data` which allows you to view the data for a specific content item including news content.

The `StartDate` and `EndDate` will default to the current date if not provided.

The data is only returned if the content data type is text based. Other content data types such as `JPEG` or `PNG` will be returned as null.

Request Details

Parameters

Name	Location	Type	Required	Description
PageSize	query	integer	No	The number of items to return for this request. Defaults to the minimum value of 10. The max value is 20.

`plaintext` 

| `LastPageKey` | query | string | No | The `LastPageKey`, where to start paged query. || `StartDate` | query | string (date-time) | No | The start effective date. This will default to today if not provided. || `EndDate` | query | string (date-time) | No | The end effective date. This will default to today if not provided. || `Type` | query | string | No | The news type to search.

Enumeration values: 0 = All, 1 = Alert, 2 = OneTimeRead, 3 = Daily, 4 = NoAlert || `RequestType` | query | string | No | The news request type.

Enumeration values: 0 = Default, 1 = Unread, 2 = UnreadAlert, 3 = UnreadDailyAndOneTimeRead || `DomainCode` | query | string | No | The domain code. See `/resources/Domains` for the valid codes. || `DepartmentCode` | query | string | No | The department code. See `/resources/Departments` for the valid codes. || `RoleCode` | query | string | No | The role code. See `/resources/Roles` for the valid codes. |

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 200

News retrieved successfully:

json



```
{
  "data": [
    {
      "newsList": [
        {
          "data": "test 1",
          "contentId": 100,
          "name": "test1",
          "description": "description1",
          "createdUserKey": "MQ--",
          "createdDate": "2024-12-02T05:01:50.2+00:00",
          "dataType": 1,
          "size": 6,
          "type": 1,
          "organizationCode": "SYSTEM",
          "domainCode": "EXT",
          "departmentCode": "TA",
          "roleCode": "TAW3",
          "effectiveDate": "2024-12-02T00:00:00",
          "discontinueDate": "2024-12-03T00:00:00",
          "retentionDate": "2024-12-04T00:00:00"
        },
        {
          "data": "{\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033{\\fonttbl{\\f0\\fnil\\fcharset0 Microsoft Sans Serif;}}\\r\\n\\viewk",
          "contentId": 101,
          "name": "test2",
          "description": "description2",
          "createdUserKey": "MQ--",
          "createdDate": "2024-12-02T05:01:50.2+00:00",
          "dataType": 2,
          "size": 138,
          "type": 2,
          "organizationCode": "SYSTEM",
          "domainCode": "EXT",
          "departmentCode": "TA",
          "roleCode": "TAW3",
          "effectiveDate": "2024-12-02T00:00:00",
          "discontinueDate": "2024-12-03T00:00:00",
          "retentionDate": "2024-12-04T00:00:00"
        },
        {
          "data": null,
          "contentId": 102,
          "name": "test3",
          "description": "description3",
          "createdUserKey": "MQ--",
          "createdDate": "2024-12-02T05:01:50.2+00:00",
          "dataType": 3,
          "size": 138,
          "type": 3,
          "organizationCode": "SYSTEM",
          "domainCode": "EXT",
          "departmentCode": "TA",
          "roleCode": "TAW3",
          "effectiveDate": "2024-12-02T00:00:00",
          "discontinueDate": "2024-12-03T00:00:00",
          "retentionDate": "2024-12-04T00:00:00"
        },
        {
          "data": null,
          "contentId": 103,
          "name": "test4",
          "description": "description4",
          "createdUserKey": "MQ--",
          "createdDate": "2024-12-02T05:01:50.2+00:00",
          "dataType": 4,
          "size": 138,
          "type": 4,
          "organizationCode": "SYSTEM",
          "domainCode": "EXT",
          "departmentCode": "TA",
          "roleCode": "TAW3",
          "effectiveDate": "2024-12-02T00:00:00",
          "discontinueDate": "2024-12-03T00:00:00",
          "retentionDate": "2024-12-04T00:00:00"
        }
      ]
    }
  ]
}
```

```
"effectiveDate": "2024-12-02T00:00:00",
"discontinueDate": "2024-12-03T00:00:00",
"retentionDate": "2024-12-04T00:00:00"
},
{
  "data": null,
  "contentId": 104,
  "name": "test5",
  "description": "description5",
  "createdUserKey": "MQ--",
  "createdDate": "2024-12-02T05:01:50.2+00:00",
  "dataType": 5,
  "size": 138,
  "type": 1,
  "organizationCode": "SYSTEM",
  "domainCode": "EXT",
  "departmentCode": "TA",
  "roleCode": "TAW3",
  "effectiveDate": "2024-12-02T00:00:00",
  "discontinueDate": "2024-12-03T00:00:00",
  "retentionDate": "2024-12-04T00:00:00"
},
{
  "data": null,
  "contentId": 105,
  "name": "test6",
  "description": "description6",
  "createdUserKey": "MQ--",
  "createdDate": "2024-12-02T05:01:50.2+00:00",
  "dataType": 6,
  "size": 138,
  "type": 2,
  "organizationCode": "SYSTEM",
  "domainCode": "EXT",
  "departmentCode": "TA",
  "roleCode": "TAW3",
  "effectiveDate": "2024-12-02T00:00:00",
  "discontinueDate": "2024-12-03T00:00:00",
  "retentionDate": "2024-12-04T00:00:00"
},
{
  "data": null,
  "contentId": 106,
  "name": "test7",
  "description": "description7",
  "createdUserKey": "MQ--",
  "createdDate": "2024-12-02T05:01:50.2+00:00",
  "dataType": 7,
  "size": 138,
  "type": 3,
  "organizationCode": "SYSTEM",
  "domainCode": "EXT",
  "departmentCode": "TA",
  "roleCode": "TAW3",
  "effectiveDate": "2024-12-02T00:00:00",
  "discontinueDate": "2024-12-03T00:00:00",
  "retentionDate": "2024-12-04T00:00:00"
},
{
  "data": "<!DOCTYPE html><html><body><h1>Digital API HTML Heading</h1><p>If you are reading this correctly, it means that t",
  "contentId": 107,
  "name": "test8",
  "description": "description8",
  "createdUserKey": "MQ--",
  "createdDate": "2024-12-02T05:01:50.2+00:00",
  "dataType": 8,
  "size": 171,
  "type": 4,
  "organizationCode": "SYSTEM",
  "domainCode": "EXT",
  "departmentCode": "TA",
  "roleCode": "TAW3",
```

```

        "effectiveDate": "2024-12-02T00:00:00",
        "discontinueDate": "2024-12-03T00:00:00",
        "retentionDate": "2024-12-04T00:00:00"
    },
    {
        "data": null,
        "contentId": 108,
        "name": "test9",
        "description": "description9",
        "createdUserKey": "MQ--",
        "createdDate": "2024-12-02T05:01:50.2+00:00",
        "dataType": 9,
        "size": 138,
        "type": 1,
        "organizationCode": "SYSTEM",
        "domainCode": "EXT",
        "departmentCode": "TA",
        "roleCode": "TAW3",
        "effectiveDate": "2024-12-02T00:00:00",
        "discontinueDate": "2024-12-03T00:00:00",
        "retentionDate": "2024-12-04T00:00:00"
    },
    {
        "data": "https://www.navitaire.com",
        "contentId": 109,
        "name": "test10",
        "description": "description10",
        "createdUserKey": "MQ--",
        "createdDate": "2024-12-02T05:01:50.2+00:00",
        "dataType": 10,
        "size": 10,
        "type": 2,
        "organizationCode": "SYSTEM",
        "domainCode": "EXT",
        "departmentCode": "TA",
        "roleCode": "TAW3",
        "effectiveDate": "2024-12-02T00:00:00",
        "discontinueDate": "2024-12-03T00:00:00",
        "retentionDate": "2024-12-04T00:00:00"
    },
    {
        "data": null,
        "contentId": 110,
        "name": "test11",
        "description": "description11",
        "createdUserKey": "MQ--",
        "createdDate": "2024-12-02T05:01:50.2+00:00",
        "dataType": 11,
        "size": 138,
        "type": 3,
        "organizationCode": "SYSTEM",
        "domainCode": "EXT",
        "departmentCode": "TA",
        "roleCode": "TAW3",
        "effectiveDate": "2024-12-02T00:00:00",
        "discontinueDate": "2024-12-03T00:00:00",
        "retentionDate": "2024-12-04T00:00:00"
    }
],
"totalCount": 11,
"lastPageKey": "MTAwMA--"
}
}

```

Description: Example when news are retrieved successfully.

No matching news item:

```
json

{
  "data": {
    "newsList": [],
    "totalCount": 0,
    "lastPageKey": "MTAwMA--"
  }
}
```

Description: Example when no news items match the search criteria.

Response 400

Retrieving news is not allowed:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:RetrieveNewsNotAllowed",
      "message": "Retrieving news is not allowed.",
      "type": "Validation",
      "details": null,
      "rawMessage": "Retrieving news is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when retrieving news is not allowed.

Retrieving content is not allowed:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:RetrieveContentNotAllowed",
      "message": "Retrieving content is not allowed.",
      "type": "Validation",
      "details": null,
      "rawMessage": "Retrieving content is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when retrieving content is not allowed.

Start date is later than the end date.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:StartDateLaterThanEndDate",
      "message": "The start effective date cannot be later than the end effective date.",
      "type": "Validation",
      "details": {
        "endDate": "10/15/2024 12:00:00 AM",
        "startDate": "10/16/2024 12:00:00 AM"
      },
      "rawMessage": "The start effective date cannot be later than the end effective date.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the start date is later than the end date.

Page size is out of range.:

```
json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "FindNewsRequest.PageSize:RangeAttribute",
      "message": "The field FindNewsRequest must be between 10 and 20.",
      "type": "Validation",
      "details": {
        "validation": "RangeAttribute",
        "model": "FindNewsRequest",
        "member": "PageSize"
      },
      "rawMessage": "The field FindNewsRequest must be between 10 and 20.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the page size is out of range. The minimum is 10 and the maximum is 20.

Last page key exceeds maximum length.:

```
json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "FindNewsRequest.LastPageKey:StringLengthAttribute",
      "message": "The field FindNewsRequest must be a string with a maximum length of 100.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "FindNewsRequest",
        "member": "LastPageKey"
      },
      "rawMessage": "The field FindNewsRequest must be a string with a maximum length of 100.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the last page key exceeds the maximum length of 100.

Domain code exceeds maximum length.:

```
json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "FindNewsRequest.DomainCode:StringLengthAttribute",
      "message": "The field FindNewsRequest must be a string with a maximum length of 3.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "FindNewsRequest",
        "member": "DomainCode"
      },
      "rawMessage": "The field FindNewsRequest must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the domain code exceeds the maximum length of 3.

Department code exceeds maximum length.:

```

json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "FindNewsRequest.DepartmentCode:StringLengthAttribute",
      "message": "The field FindNewsRequest must be a string with a maximum length of 4.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "FindNewsRequest",
        "member": "DepartmentCode"
      },
      "rawMessage": "The field FindNewsRequest must be a string with a maximum length of 4.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the department code exceeds the maximum length of 4.

Role code exceeds maximum length.:.

```

json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "FindNewsRequest.RoleCode:StringLengthAttribute",
      "message": "The field FindNewsRequest must be a string with a maximum length of 4.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "FindNewsRequest",
        "member": "RoleCode"
      },
      "rawMessage": "The field FindNewsRequest must be a string with a maximum length of 4.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the role code exceeds the maximum length of 4.

POST /api/nsk/v1/contents/news

Summary: Add a new news content item.

Description

Add news or sometimes referred to as FYI Content notes.

Related endpoints:

- GET /resources/contents/{contentID}/data : Allows you to view the data for a specific content item including news content.
- GET /contents/news : Allows you to search for news content items.

The `OrganizationCode` is automatically set to the `OrganizationCode` of the logged in agent if the `DomainCode`, `DepartmentCode`, and `RoleCode` are all set in the request.

Only text based content data types are allowed. Other content data types such as `JPEG` or `PNG` must be stored elsewhere.

Request Details

Request Body

The add news request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

400 Response

There was a problem with the request and it could not be completed.

Examples

Request

Request with content data type set to text.:

```
json   
{  
    "name": "NewsWithText",  
    "description": "News with a text data",  
    "type": 1,  
    "effectiveDate": "2024-10-15T00:00:00",  
    "discontinueDate": "2024-10-16T00:00:00",  
    "retentionDate": "2024-10-17T00:00:00",  
    "dataType": 1,  
    "data": "Sample text file for Digital API Testing.",  
    "domainCode": "",  
    "departmentCode": "",  
    "roleCode": ""  
}
```

Description: Example when the content data type is set to text.

Request with content data type set to rich text format.:

```
json   
{  
    "name": "NewsWithRichTextFormat",  
    "description": "News with a rich text format data",  
    "type": 2,  

```

Description: Example when the content data type is set to rich text format.

Request with content data type set to HTML.:

```
json   
{  
    "name": "NewsWithHtml",  
    "description": "News with a html data",  
    "type": 3,  
    "effectiveDate": "2024-10-15T00:00:00",  
    "discontinueDate": "2024-10-16T00:00:00",  
    "retentionDate": "2024-10-17T00:00:00",  
    "dataType": 8,  
    "data": "<!DOCTYPE html><html><body><h1>Digital API HTML Heading</h1><p>If you are reading this correctly, it means that the new  
    "domainCode": "",  
    "departmentCode": "",  
    "roleCode": ""  
}
```

Description: Example when the content data type is set to HTML.

Request with content data type set to URI.:

```
json   
{  
    "name": "NewsWithUri",  
    "description": "News with an Uri data",  
    "type": 4,  
    "effectiveDate": "2024-10-15T00:00:00",  
    "discontinueDate": "2024-10-16T00:00:00",  
    "retentionDate": "2024-10-17T00:00:00",  
    "dataType": 10,  
    "data": "https://www.navitaire.com",  
    "domainCode": "",  
    "departmentCode": "",  
    "roleCode": ""  
}
```

Description: Example when the content data type is set to URI.

Request with no discontinue date and retention date.:

```
json   
{  
    "name": "test1",  
    "description": "description",  
    "type": 1,  
    "effectiveDate": "2024-10-15T00:00:00",  
    "discontinueDate": null,  
    "retentionDate": null,  
    "dataType": 1,  
    "data": "text",  
    "domainCode": "",  
    "departmentCode": "",  
    "roleCode": ""  
}
```

Description: Example when the discontinue date and the retention date is not set.

News added successfully with organization code automatically set.:

```
json
```

```
{  
    "name": "test1",  
    "description": "description",  
    "type": 1,  
    "effectiveDate": "2024-10-15T00:00:00",  
    "discontinueDate": "2024-10-16T00:00:00",  
    "retentionDate": "2024-10-17T00:00:00",  
    "dataType": 1,  
    "data": "text",  
    "domainCode": "EXT",  
    "departmentCode": "TA",  
    "roleCode": "TAW3"  
}
```

Description: Example when news is added successfully with the organization code automatically set to the logged in agent's organization.

Response 201

News with text data type is added successfully.:

```
json
```

```
{  
    "data": {  
        "news": {  
            "contentId": 240,  
            "name": "NewsWithText",  
            "description": "News with a text data",  
            "createdUserKey": "MTE5MDY-",  
            "createdDate": "2024-10-15T02:08:37.69+00:00",  
            "dataType": 1,  
            "size": 41,  
            "type": 1,  
            "organizationCode": null,  
            "domainCode": null,  
            "departmentCode": null,  
            "roleCode": null,  
            "effectiveDate": "2024-10-15T00:00:00",  
            "discontinueDate": "2024-10-16T00:00:00",  
            "retentionDate": "2024-10-17T00:00:00"  
        }  
    }  
}
```

Description: Example when a news with a text data type is added successfully.

News with rich text format data type is added successfully.:

```
json

{
  "data": {
    "news": {
      "contentId": 240,
      "name": "NewsWithRichTextFormat",
      "description": "News with a rich text format data",
      "createdUserKey": "MTE5MDY-",
      "createdDate": "2024-10-15T02:08:37.69+00:00",
      "dataType": 2,
      "size": 160,
      "type": 2,
      "organizationCode": null,
      "domainCode": null,
      "departmentCode": null,
      "roleCode": null,
      "effectiveDate": "2024-10-15T00:00:00",
      "discontinueDate": "2024-10-16T00:00:00",
      "retentionDate": "2024-10-17T00:00:00"
    }
  }
}
```

Description: Example when a news with a rich text format data type is added successfully.

News with HTML data type is added successfully.:

```
json

{
  "data": {
    "news": {
      "contentId": 240,
      "name": "NewsWithHtml",
      "description": "News with a html data",
      "createdUserKey": "MTE5MDY-",
      "createdDate": "2024-10-15T02:08:37.69+00:00",
      "dataType": 8,
      "size": 171,
      "type": 3,
      "organizationCode": null,
      "domainCode": null,
      "departmentCode": null,
      "roleCode": null,
      "effectiveDate": "2024-10-15T00:00:00",
      "discontinueDate": "2024-10-16T00:00:00",
      "retentionDate": "2024-10-17T00:00:00"
    }
  }
}
```

Description: Example when a news with a HTML data type is added successfully.

News with URI data type is added successfully.:

```
json

{
  "data": {
    "news": {
      "contentId": 240,
      "name": "NewsWithUri",
      "description": "News with an Uri data",
      "createdUserKey": "MTE5MDY-",
      "createdDate": "2024-10-15T02:08:37.69+00:00",
      "dataType": 10,
      "size": 25,
      "type": 4,
      "organizationCode": null,
      "domainCode": null,
      "departmentCode": null,
      "roleCode": null,
      "effectiveDate": "2024-10-15T00:00:00",
      "discontinueDate": "2024-10-16T00:00:00",
      "retentionDate": "2024-10-17T00:00:00"
    }
  }
}
```

Description: Example when a news with an URI data type is added successfully.

News added successfully with no discontinue date and retention date.:

```
json

{
  "data": {
    "news": {
      "contentId": 241,
      "name": "test1",
      "description": "description",
      "createdUserKey": "MTE5MDY-",
      "createdDate": "2024-10-15T02:09:06.263+00:00",
      "dataType": 1,
      "size": 3,
      "type": 1,
      "organizationCode": null,
      "domainCode": null,
      "departmentCode": null,
      "roleCode": null,
      "effectiveDate": "2024-10-15T00:00:00",
      "discontinueDate": null,
      "retentionDate": null
    }
  }
}
```

Description: Example when news is added with no discontinue date and retention date.

News added successfully with organization code automatically set.:

```
json   
{  
  "data": {  
    "news": {  
      "contentId": 242,  
      "name": "test1",  
      "description": "description",  
      "createdUserKey": "MTE5MDY-",  
      "createdDate": "2024-10-15T02:27:43.83+00:00",  
      "dataType": 1,  
      "size": 3,  
      "type": 1,  
      "organizationCode": "88887777",  
      "domainCode": "EXT",  
      "departmentCode": "TA",  
      "roleCode": "TAW3",  
      "effectiveDate": "2024-10-15T00:00:00",  
      "discontinueDate": "2024-10-16T00:00:00",  
      "retentionDate": "2024-10-17T00:00:00"  
    }  
  }  
}
```

Description: Example when news is added successfully with the organization code automatically set to the logged in agent's organization.

Response 400

Name is null or empty.:

```
json   
{  
  "errors": [  
    {  
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",  
      "code": "AddNewsRequest.Name:RequiredAttribute",  
      "message": "AddNewsRequest.Name:RequiredAttribute",  
      "type": "Validation",  
      "details": {  
        "member": "Name",  
        "model": "AddNewsRequest",  
        "validation": "RequiredAttribute"  
      },  

```

Description: Example when the name is null or empty.

News type is null.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "AddNewsRequest.Type:RequiredAttribute",
      "message": "AddNewsRequest.Type:RequiredAttribute",
      "type": "Validation",
      "details": {
        "member": "Type",
        "model": "AddNewsRequest",
        "validation": "RequiredAttribute"
      },
      "rawMessage": "The AddNewsRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the news type is null.

Content data type is null.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "AddNewsRequest.DataType:RequiredAttribute",
      "message": "AddNewsRequest.DataType:RequiredAttribute",
      "type": "Validation",
      "details": {
        "member": "DataType",
        "model": "AddNewsRequest",
        "validation": "RequiredAttribute"
      },
      "rawMessage": "The AddNewsRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the content data type is null.

Data is null or empty.:

```

json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "AddNewsRequest.Data:RequiredAttribute",
      "message": "AddNewsRequest.Data:RequiredAttribute",
      "type": "Validation",
      "details": {
        "member": "Data",
        "model": "AddNewsRequest",
        "validation": "RequiredAttribute"
      },
      "rawMessage": "The AddNewsRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the data is null or empty.

Effective date is equal to the discontinuedate.:

```

json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:EffectiveDateCannotBeEqualToDiscontinueDate",
      "message": "The effective date cannot be the same as the discontinue date.",
      "type": "Validation",
      "details": {
        "discontinueDate": "10/15/2024 12:00:00 AM",
        "effectiveDate": "10/15/2024 12:00:00 AM"
      },
      "rawMessage": "The effective date cannot be the same as the discontinue date.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the effective date is equal to the discontinue date.

EffectiveDate is later than the DiscontinueDate.:

```

json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "News.DiscontinueDate:MinMaxDateTimeAttribute",
      "message": "The value of News.DiscontinueDate:MinMaxDateTimeAttribute is 10/15/2024 00:00:00 but must be between 10/16/2024 00:00:00 and 10/15/2024 23:59:59",
      "type": "Validation",
      "details": null,
      "rawMessage": "The value of News.DiscontinueDate:MinMaxDateTimeAttribute is 10/15/2024 00:00:00 but must be between 10/16/2024 00:00:00 and 10/15/2024 23:59:59",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the effective date is later than the discontinue date.

Discontinue date is later than the retention date.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "News.RetentionDate:MinMaxDateTimeAttribute",
      "message": "The value of News.RetentionDate:MinMaxDateTimeAttribute is 10/16/2024 00:00:00 but must be between 10/17/2024 00:00:00 and 10/17/2024 12:00:00.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The value of News.RetentionDate:MinMaxDateTimeAttribute is 10/16/2024 00:00:00 but must be between 10/17/2024 00:00:00 and 10/17/2024 12:00:00.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the discontinue date is later than the retention date.

Discontinue date is null and the retention date is not null.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:RetentionDateCannotBeSetIfDiscontinueDateIsNull",
      "message": "Retention date cannot be set if discontinue date is null.",
      "type": "Validation",
      "details": {
        "discontinueDate": "",
        "retentionDate": "10/17/2024 12:00:00 AM"
      },
      "rawMessage": "Retention date cannot be set if discontinue date is null.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the discontinue date is null and the retention date is not null.

News type is invalid.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:InvalidNewsType",
      "message": "Invalid news type.",
      "type": "Validation",
      "details": {
        "newsType": "10"
      },
      "rawMessage": "Invalid news type.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the news type is invalid.

Content data type is invalid.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:InvalidContentDataType",
      "message": "Invalid content data type. Allowed values are 'Text', 'RichTextFormat', 'Html', and 'Uri'.",
      "type": "Validation",
      "details": {
        "contentDataType": "20"
      },
      "rawMessage": "Invalid content data type. Allowed values are 'Text', 'RichTextFormat', 'Html', and 'Uri'.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the Content data type is invalid.

Domain code is invalid.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:InvalidDomainCode",
      "message": "Invalid domain code.",
      "type": "Validation",
      "details": {
        "domainCode": "abc"
      },
      "rawMessage": "Invalid domain code.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the domain code is invalid. See /resources/Domains for valid codes.

Department code is invalid.:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:InvalidDepartmentCode",
      "message": "Invalid department code.",
      "type": "Validation",
      "details": {
        "departmentCode": "abc"
      },
      "rawMessage": "Invalid department code.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example when the department code is invalid. See /resources/Departments for valid codes.

Role code is invalid.:

```
json   
{  
  "errors": [  
    {  
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",  
      "code": "nsk:News:InvalidRoleCode",  
      "message": "Invalid role code.",  
      "type": "Validation",  
      "details": {  
        "roleCode": "abc"  
      },  
      "rawMessage": "Invalid role code.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    },  
  ],  
  "data": null  
}
```

Description: Example when the role code is invalid. See /resources/Roles for valid codes.

Domain code exceeds maximum length.:

```
json   
{  
  "errors": [  
    {  
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",  
      "code": "AddNewsRequest.DomainCode:StringLengthAttribute",  
      "message": "The field AddNewsRequest must be a string with a maximum length of 3.",  
      "type": "Validation",  
      "details": {  
        "validation": "StringLengthAttribute",  
        "model": "AddNewsRequest",  
        "member": "DomainCode"  
      },  
      "rawMessage": "The field AddNewsRequest must be a string with a maximum length of 3.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    },  
  ],  
  "data": null  
}
```

Description: Example when the domain code exceeds the maximum length of 3.

Department code exceeds maximum length.:

```

json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "AddNewsRequest.DepartmentCode:StringLengthAttribute",
      "message": "The field AddNewsRequest must be a string with a maximum length of 4.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "AddNewsRequest",
        "member": "DepartmentCode"
      },
      "rawMessage": "The field AddNewsRequest must be a string with a maximum length of 4.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the department code exceeds the maximum length of 4.

Role code exceeds maximum length.:

```

json

{
  "errors": [
    {
      "id": "b79d85c8-b090-baa6-5efa-dbc168844eeb",
      "code": "AddNewsRequest.RoleCode:StringLengthAttribute",
      "message": "The field AddNewsRequest must be a string with a maximum length of 4.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "AddNewsRequest",
        "member": "RoleCode"
      },
      "rawMessage": "The field AddNewsRequest must be a string with a maximum length of 4.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when the role code exceeds the maximum length of 4.

Adding news is not allowed.:

```

json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:News:AddNewsNotAllowed",
      "message": "Adding news is not allowed.",
      "type": "Validation",
      "details": null,
      "rawMessage": "Adding news is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example when adding news is not allowed.

Currency Endpoints

GET /api/nsk/v1/currency/converter

Summary: Calculates currency conversions.

Description

If the currency conversion is inverted, the exchange rate returned is of the reversed to and from currency code and the rate used in the calculation is 1/(exchange rate of the reverse).

Request Details

Parameters

Name	Location	Type	Required	Description
FromCurrencyCode	query	string	No	The currency code that the amount is currently in.

```
plaintext   
|
```

| ToCurrencyCode | query | string | No | The currency code we want the amount converted to. || Amount | query | number (decimal) | No | The amount to be converted. This will be in the FromCurrencyCode currency. || Inverted | query | boolean | No | If set to true, the endpoint will use 1/(exchange rate of the reverse) during the conversion. Defaults to false if no value is set. |

Responses

200 Response

400 Response

Examples

Response 200

Direct currency conversion successful:

```
json   
{  
  "data": {  
    "convertedAmount": 2319.4,  
    "fromCurrencyCode": "EUR",  
    "exchangeRate": 1.16,  
    "toCurrencyCode": "USD",  
    "roundingFactor": 0.1,  
    "amount": 1999.52,  
    "marketingRoundingFactor": 0.1,  
    "inverted": false  
  }  
}
```

Description: Example when using direct currency conversion.

Inverse currency conversion successful:

```
json

{
  "data": {
    "convertedAmount": 199.5,
    "fromCurrencyCode": "EUR",
    "exchangeRate": 1.16,
    "toCurrencyCode": "USD",
    "roundingFactor": 0.1,
    "amount": 231.4,
    "marketingRoundingFactor": 0.1,
    "inverted": true
  }
}
```

Description: Example when using inverse currency conversion.

Response 400

Invalid Currency Code Error:

```
json

{
  "errors": [
    {
      "id": "fc79d927-e7c3-91ce-8ae8-fbd0adf7bb39",
      "code": "nsk:Currency:InvalidCurrencyCode",
      "message": "nsk:Currency:InvalidCurrencyCode",
      "type": "Validation",
      "details": null,
      "rawMessage": "The currency code ABC is either invalid or inactive in the current environment.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when currency code is invalid.

Currency Conversion Failed:

```
json

{
  "errors": [
    {
      "id": "1a68ef3e-44b5-08d4-9395-e69695467a5d",
      "code": "nsk:Currency:CurrencyConversionFailed",
      "message": "nsk:Currency:CurrencyConversionFailed",
      "type": "Validation",
      "details": null,
      "rawMessage": "The currency conversion failed unexpectedly. Please verify the currency codes are valid and active in the current environment.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when currency conversion failed.

Amount Required Error:

```
json

{
  "errors": [
    {
      "id": "e53dd703-1362-d1b2-64c4-c9b1a5f853fb",
      "code": "CurrencyConversionRequest.Amount:RequiredAttribute",
      "message": "CurrencyConversionRequest.Amount:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "CurrencyConversionRequest",
        "member": "Amount"
      },
      "rawMessage": "The CurrencyConversionRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when amount is not set.

ToCurrencyCode Required Error:

```
json

{
  "errors": [
    {
      "id": "c20fe6e2-2d91-7855-1d0b-613f97da120e",
      "code": "CurrencyConversionRequest.ToCurrencyCode:RequiredAttribute",
      "message": "CurrencyConversionRequest.ToCurrencyCode:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "CurrencyConversionRequest",
        "member": "ToCurrencyCode"
      },
      "rawMessage": "The CurrencyConversionRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when ToCurrencyCode is not set.

FromCurrencyCode Required Error:

```

json

{
  "errors": [
    {
      "id": "1b6d7366-8ae6-b361-8333-f60f9c7306ac",
      "code": "CurrencyConversionRequest.FromCurrencyCode:RequiredAttribute",
      "message": "CurrencyConversionRequest.FromCurrencyCode:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "CurrencyConversionRequest",
        "member": "FromCurrencyCode"
      },
      "rawMessage": "The CurrencyConversionRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example error when FromCurrencyCode is not set.

Equipment Endpoints

POST /api/dcs/v1/equipment/swap

Summary: Exchanges one equipment for another on an existing flight.

Description

Specifically, the leg is updated with a new transport code and equipment information. All seat assignments on the affected flight legs are examined. Any assignments for seats which do not exist in the new equipment are released. Those passengers will no longer have seat assignments on the affected legs. As a result of the equipment swap, passengers may end up with a different cabin or class of service than they had on the prior equipment. This method is only concerned with making sure that the seat assignment physically exists on the new configuration. Any seat holds or blocks on seats which no longer exist are also released. This method does not revise the inventory or SSR lids on the swapped flight legs. These values may become invalid after an equipment swap. This method also may return a list of passengers what were moved from one equipment to another that had a seat change.

GraphQL endpoints: equipmentSwap, equipmentSwapBatch

Request Details

Request Body

The equipment swap request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

POST /api/dcs/v1/equipment/tailNumber

Summary: Assigns a tail number to a specific leg.

Description

To confirm the assignment of the tail number, please use /api/nsk/v2/trip/info/{legKey}/status. If the aircraft code of the leg is different from the aircraft code of the tail number, an equipment swap will be required to assign the tail number and the tail number will not be changed until that swap. This equipment swap requirement can be overridden using `IgnoreEquipmentDifference` on the request, however this should be used with caution.

GraphQL endpoints: tailNumber, tailNumberBatch

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

404 Response

Etickets Endpoints

GET /api/nsk/v2/eTickets/{eTicketNumber}

Summary: Retrieves the passenger's e-ticket and its details by providing the ticket number, hosted carrier code and the alternate carrier code from the e-ticket database if the ticket is OA validated (issued by the other airline not by the hosted carrier).

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTicketNumber	path	string	Yes	The e-ticket number.
HostedCarrierCode	query	string	No	The hosted carrier code. If not provided, the hosted carrier code will be retrieved internally.

```
plaintext
|
|
```

| AlternateCarrierCode | query | string | No | The alternate carrier code. This is required if the ticket is issued by another airline. |

Responses

200 Response

400 Response

404 Response

Examples

Response 200

Initially issued e-tickets:

json



```
{
  "eTicket": {
    "passengerName": {
      "first": "Example",
      "middle": null,
      "last": "Passenger",
      "title": null,
      "suffix": null
    },
    "connectedETickets": [
      {
        "couponCount": 1,
        "ticketNumber": "0286010103190",
        "ticketEmds": null
      }
    ],
    "exchangedETickets": [],
    "recordLocators": [
      {
        "recordLocatorKey": "Vzk0SFVHIQa--",
        "recordCode": "ABCDEF",
        "systemDomainCode": null,
        "owningSystemCode": null,
        "bookingSystemCode": "NV",
        "interactionPurpose": null,
        "hostedCarrierCode": null
      }
    ],
    "fareCalculation": "SLC NV BOS84.65USD84.65END",
    "issueDate": "2025-08-07T00:00:00Z",
    "companyIdNumber": "0000",
    "chargeCode": null,
    "totalTaxAmount": 25.0,
    "taxDetailsList": [
      {
        "taxCode": "CGY",
        "amount": 20.0,
        "currencyCode": "USD"
      }
    ],
    "ticketFare": {
      "baseAmount": 150.0,
      "currencyCode": "USD",
      "foreignAmount": 0.0,
      "foreignCurrencyCode": null
    },
    "totalAmount": 200.0,
    "totalAmountCurrencyCode": "USD",
    "remarks": [
      {
        "textQualifier": 3,
        "informationType": 10,
        "freeTexts": [
          "THIS COMMENT IS FOR VOLUNTARY ISSUANCE OF E TICKET"
        ]
      },
      {
        "textQualifier": 3,
        "informationType": 12,
        "freeTexts": [
          "SLC NV BOS84.65USD84.65END"
        ]
      },
      {
        "textQualifier": 3,
        "informationType": 13,
        "freeTexts": [
          "SALT LAKE CITY US",
          "NAV1 DEFAULT CARRIER"
        ]
      }
    ],
  }
}
```

```
{  
    "textQualifier": 3,  
    "informationType": 14,  
    "freeTexts": [  
        "SLC10NOV231234567"  
    ]  
},  
    "createdUserIdentifier": "45321"  
},  
    "status": 1,  
    "alternateCarrierCode": "NV"  
}
```

Description: Example response for successful retrieval of issued e-tickets

Re-issued e-tickets:

json



```
{
  "eTicket": {
    "passengerName": {
      "first": "Example",
      "middle": null,
      "last": "Passenger",
      "title": null,
      "suffix": null
    },
    "connectedETickets": [
      {
        "couponCount": 1,
        "ticketNumber": "0286010103191",
        "ticketEmds": null
      }
    ],
    "exchangedETickets": [
      {
        "couponCount": 1,
        "ticketNumber": "0286010103190",
        "ticketEmds": null
      }
    ],
    "recordLocators": [
      {
        "recordLocatorKey": "Vzk0SFVHIQa--",
        "recordCode": "ABCDEF",
        "systemDomainCode": null,
        "owningSystemCode": null,
        "bookingSystemCode": "NV",
        "interactionPurpose": null,
        "hostedCarrierCode": null
      }
    ],
    "fareCalculation": "SLC NV BOS84.65USD84.65END",
    "issueDate": "2025-08-07T00:00:00Z",
    "companyIdNumber": "0000",
    "chargeCode": null,
    "totalTaxAmount": 25.0,
    "taxDetailsList": [
      {
        "taxCode": "CGY",
        "amount": 20.0,
        "currencyCode": "USD"
      }
    ],
    "ticketFare": {
      "baseAmount": 150.0,
      "currencyCode": "USD",
      "foreignAmount": 0.0,
      "foreignCurrencyCode": null
    },
    "totalAmount": 200.0,
    "totalAmountCurrencyCode": "USD",
    "remarks": [
      {
        "textQualifier": 3,
        "informationType": 10,
        "freeTexts": [
          "THIS COMMENT IS FOR INVOLUNTARY ISSUANCE OF E TICKET"
        ]
      },
      {
        "textQualifier": 3,
        "informationType": 12,
        "freeTexts": [
          "SLC NV BOS84.65USD84.65END"
        ]
      },
      {
        "textQualifier": 3,
        "informationType": 14,
        "freeTexts": [
          "THIS COMMENT IS FOR INVOLUNTARY ISSUANCE OF E TICKET"
        ]
      }
    ]
  }
}
```

```

    "informationType": 13,
    "freeTexts": [
        "SALT LAKE CITY US",
        "NAVI DEFAULT CARRIER"
    ],
},
{
    "textQualifier": 3,
    "informationType": 14,
    "freeTexts": [
        " 0286010103190  SLC10NOV231234567"
    ]
},
],
"createdUserIdentifer": "45321"
},
"status": 1,
"alternateCarrierCode": "NV"
}

```

Description: Example response for successful retrieval of re-issued e-tickets

Response 400

Non-existent e-Ticket:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:ETicket:ETicketNotFoundInETicketDatabase",
      "message": "nsk:ETicket:ETicketNotFoundInETicketDatabase",
      "type": "Error",
      "details": null,
      "rawMessage": "The e-ticket number 0286010103199 was not found in the e-ticket database.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when retrieving non-existent or invalid e-tickets

GET /api/nsk/v1/eTickets/{eTicketNumber}/coupons

Summary: Retrieves all coupons associated with the e-ticket.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTicketNumber	path	string	Yes	The e-ticket number.
HostedCarrierCode	query	string	No	The hosted carrier code. If not provided, the hosted carrier code will be retrieved internally.

```
plaintext
|
|
```

| AlternateCarrierCode | query | string | No | The alternate carrier code. This is required if the ticket is issued by another airline. |

Responses

200 Response

404 Response

GET /api/nsk/v1/eTickets/{eTicketNumber}/coupons/{couponNumber}

Summary: Retrieves a single coupon associated with the e-ticket.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTicketNumber	path	string	Yes	The e-ticket number.
couponNumber	path	integer	Yes	The e-ticket coupon number.
HostedCarrierCode	query	string	No	The hosted carrier code. If not provided, the hosted carrier code will be retrieved internally.

plaintext 

| AlternateCarrierCode | query | string | No | The alternate carrier code. This is required if the ticket is issued by another airline. |

Responses

200 Response

404 Response

PUT /api/nsk/v2/eTickets/{eTicketNumber}/coupons/{couponNumber}

Summary: Updates a coupon contained in an e-ticket.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTicketNumber	path	string	Yes	The e-ticket number to update.
couponNumber	path	integer	Yes	The coupon number to update

Request Body

The e-ticket coupon update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

Farerules Endpoints

GET /api/nsk/v1/fareRules/{fareAvailabilityKey}

Summary: Gets the fare rules from the server based on a fare availability key.

Description

Note that this defaults to returning the fare rules in html format. The html format may not be entirely culture sensitive, so setting the convertToHtml flag to false will allow users some flexibility when converting the content and the fare rules will be returned as an rtf string.

Request Details

Parameters

Name	Location	Type	Required	Description
fareAvailabilityKey	path	string	Yes	The unique fare availability key to get rules for.
convertToHtml	query	boolean	No	Whether to convert to html. Defaults to true.

Responses

200 Response

The fare rules.

404 Response

POST /api/nsk/v2/fareRules/category50/journeys/{journeyKey}/segments/{segmentKey}

Summary: Get the category 50 fare rule information after an availability call.

Description

This is functionally a GET but POST is used to reduce URL length overhead.

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
segmentKey	path	string	Yes	The segment key.

Request Body

The v2 category 50 request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

Graph Endpoints

POST /api/v2/graph/{queryName}

Summary: Invokes a graph query and returns the response.

Description

Invokes a graph query and returns the response.

Request Details

Parameters

Name	Location	Type	Required	Description
queryName	path	string	Yes	The unique name used for reporting.
cachedResults	query	boolean	No	Cached results query string parameter. This is the same as the post variable but enables various tools to set the same value.

```
plaintext
```



|

Request Body

Query details.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

Examples

Request

Simple Graph Request:

```
json
```



```
{
  "query": "{tokenKeepAlive}",
  "variables": null,
  "cachedResults": false
}
```

Description: Its recommended that you use a graph specific tool for this endpoint. See the developer documentation for more information.

Response 200

Simple Response:

```
json
```



```
{
  "data": {
    "tokenKeepAlive": true
  }
}
```

Description: The response will be json and dynamic to your graph query.

POST /api/nsk/v1/graph/{queryName}

Summary: Invokes a graph query configured in utilities.

Description

When a graph query is configured on the server under Resource Management it allows your payload sizes to be even smaller.

Request Details

Parameters

Name	Location	Type	Required	Description
queryName	path	string	Yes	The query name matching your resource management configuration.
cachedResults	query	boolean	No	True to return cached results.

Request Body

The json matching your json variables in your graph query.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The graph json response.

Health Endpoints

GET /api/v3/health

Summary: Performs the health check.

Description

This endpoint does not require a token.

plaintext



If the health check returns an overall status of "Ok", the expected HTTP status code response would be a "200 - Ok".

If the health check returns an overall status of "Warning", the expected HTTP status code response would be a "299". This is a custom code not typically mapped as per industry standards, and was done intentionally to indicate a warning response from the health endpoint.

If the health check returns an overall status of "Error", the expected HTTP status code response would be a "500 - InternalServerError".

Request Details

Responses

200 Response

OK. Request completed successfully.

299 Response

Request completed successfully with warnings from health endpoint.

500 Response

Request to health endpoint did not complete successfully.

Inventory Endpoints

GET /api/dcs/v2/inventory/legs/{legKey}

Summary: Gets the stateless inventory leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

404 Response

PATCH /api/dcs/v2/inventory/legs/{legKey}

Summary: Patches an inventory leg's data, its nests' data, and its classes' data based on information in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Request Body

The inventory leg edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

Examples

Request

Patching leg, nest, and class:

```
json   
  
{  
    "lid": 50,  
    "adjustedCapacity": 48,  
    "sendAvsMessages": true,  
    "status": 0,  
    "nests": {  
        "mtEzNTk2MTAxNSE0IU4-": {  
            "lid": 24,  
            "adjustedCapacity": 23,  
            "classes": null  
        },  
        "mtEzNTk2MTAxNSExIU4-": {  
            "lid": null,  
            "adjustedCapacity": null,  
            "classes": {  
                "mtEzNTk2MTAxNSExIU0-": {  
                    "type": 0,  
                    "rank": 1,  
                    "authorizedUnits": 10,  
                    "allotted": 9,  
                    "latestAdvancedReservation": 0  
                }  
            }  
        },  
        "mtEzNTk2MTAxNSEyIU4-": {  
            "lid": 50,  
            "adjustedCapacity": 48,  
            "classes": {  
                "mtEzNTk2MTAxNSEyIU4-": {  
                    "type": null,  
                    "rank": null,  
                    "authorizedUnits": 10,  
                    "allotted": null,  
                    "latestAdvancedReservation": null  
                },  
                "mtEzNTk2MTAxNSEyIU8-": {  
                    "type": null,  
                    "rank": null,  
                    "authorizedUnits": null,  
                    "allotted": 36,  
                    "latestAdvancedReservation": 5  
                }  
            }  
        }  
    }  
}
```

Description: Example request for patching inventory leg, nest, and class data

Patching just leg lid:

```
json   
  
{  
    "lid": 50,  
    "adjustedCapacity": null,  
    "sendAvsMessages": false,  
    "status": null,  
    "nests": null  
}
```

Description: Example request for patching just inventory leg lid

Patching leg class:

```

json

{
    "lid": null,
    "adjustedCapacity": null,
    "sendAvsMessages": false,
    "status": null,
    "nests": {
        "mtEzNTk2MTAxNSExIU4-": {
            "lid": null,
            "adjustedCapacity": null,
            "classes": {
                "mtEzNTk2MTAxNSExIU0-": {
                    "type": null,
                    "rank": null,
                    "authorizedUnits": null,
                    "allotted": 9,
                    "latestAdvancedReservation": null
                }
            }
        }
    }
}

```

Description: Example request for patching just an inventory class

GET /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}

Summary: Gets an inventory leg nest by key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
legNestKey	path	string	Yes	The leg nest key.

Responses

200 Response

404 Response

PATCH /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}

Summary: Patches an inventory leg nest's data based on information in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
legNestKey	path	string	Yes	The leg nest key.

Request Body

The inventory nest edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

GET /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}/classes/{legClassKey}

Summary: Gets an inventory leg nest class by key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
legNestKey	path	string	Yes	The leg nest key.
legClassKey	path	string	Yes	The leg class key.

Responses

200 Response

404 Response

PATCH /api/dcs/v1/inventory/legs/{legKey}/nests/{legNestKey}/classes/{legClassKey}

Summary: Patches an inventory leg nest class's data based on information in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
legNestKey	path	string	Yes	The leg nest key.
legClassKey	path	string	Yes	The leg class key.

Request Body

The inventory class edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

GET /api/dcs/v1/inventory/routes/{tripKey}/classes

Summary: Gets the classes for a route, stateless.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
tripKey	path	string	Yes	The trip key from a manifest or flight info.

Responses

200 Response

404 Response

PATCH /api/dcs/v1/inventory/routes/{tripKey}/classes/{classOfService}

Summary: Patches an inventory route class's data based on information in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
tripKey	path	string	Yes	The trip key.
classOfService	path	string	Yes	The class of service.

Request Body

The inventory class edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/dcs/v1/inventory/routes/{tripKey}/classes/{classOfService}/authorizedUnit

Summary: Deletes an inventory route class's authorized unit.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
tripKey	path	string	Yes	The trip key.
classOfService	path	string	Yes	The class of service.

Responses

200 Response

Manifest Endpoints

GET /api/nsk/v2/manifest

Summary: Gets the manifest trip information response list for the specified manifest trip information search request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Origin	query	string	No	The departure station.

```
plaintext
```

| Destination | query | string | No | The arrival station. || CarrierCode | query | string | No | The carrier code. || BeginDate | query | string (date-time) | No | The starting date of the trip information search. || Identifier | query | string | No | The transportation identifier, such as a flight number. || FlightType | query | string | No | The flight type.

Enumeration values: 0 = All, 1 = NonStop, 2 = Through, 3 = Direct, 4 = Connect |

Responses

404 Response

200 Response

The manifest trip information search results model.

GET /api/nsk/v1/manifest/{legKey}

Summary: Gets the manifest for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.

Responses

404 Response

200 Response

The manifest model.

Examples

Response 200

Manifest Light Response:

json



```
{
  "data": {
    "passengers": [
      {
        "balanceDue": true,
        "userKey": "Nzk3MTZ-",
        "personKey": "MA--",
        "passengerAlternateKey": "NML4Tgw",
        "passengerKey": "MCFBRFQ-",
        "checkInType": 0,
        "customerNumber": "888888888",
        "discountCode": null,
        "infant": null,
        "name": {
          "first": "First",
          "middle": "Middle",
          "last": "Last",
          "title": "MR",
          "suffix": null
        },
        "paidStatus": 0,
        "profileStatus": 0,
        "program": null,
        "recordLocator": "R35N2F",
        "sourcePointOfSale": {
          "agentCode": "agent",
          "domainCode": "EXT",
          "locationCode": "WWW",
          "organizationCode": "88887777"
        },
        "bookingStatus": 0,
        "passengerInformation": {
          "nationality": "US",
          "residentCountry": "US",
          "gender": 1,
          "dateOfBirth": "2001-01-01T00:00:00",
          "familyNumber": null
        },
        "type": "ADT",
        "bookingDate": "2025-08-07T04:47:17.092301+00:00",
        "soldDate": "2025-08-07T04:47:17.0923028+00:00",
        "journey": {
          "customerRecognition": {
            "code": null,
            "points": 0
          },
          "segmentCount": 1,
          "segment": {
            "segmentKey": null,
            "bookingStatus": "HK",
            "changeReason": 0,
            "classOfService": "N",
            "fareBasisCode": "NNNN",
            "leg": {
              "boardingSequence": 0,
              "boardingZone": 3,
              "compartmentDesignator": "E",
              "legNumber": 1,
              "liftStatus": 0,
              "priorityCode": null,
              "priorityDate": "2025-08-07T04:47:17.0923069+00:00",
              "unitAssignmentWeight": 0,
              "unitDesignator": "1D",
              "manifestLegKey": "NjM4ODIzNDU2MDAwMDAwMDAwIUsWITU2MjUhICFRU4hU0xD"
            },
            "productClassCode": "EC",
            "seatProperties": [],
            "soldDate": "2025-08-07T04:47:17.0923101+00:00",
            "ticketIndicator": null,
            "ticketNumber": null,
            "upgradeIndicator": false,
            "travelerIndex": 1
          }
        }
      }
    ]
  }
}
```

```

        "externalIdentifier": null,
        "legCount": 1,
        "segmentType": 0,
        "manifestSegmentKey": "NjM4ODIzNDU2MDAwMDAwMDAwIu5WITU2MjUhICFERU4hU0xD"
    },
    "changeReason": 0
}
],
"cabinPassengerCounts": [
{
    "classRank": 0,
    "checkedIn": 0,
    "cabinOfService": "F",
    "clearedStandBy": 0,
    "clearedThruStandby": 0,
    "manifested": 0,
    "noShow": 0,
    "standBy": 0,
    "thruCheckedIn": 0,
    "thruManifested": 0,
    "thruNoShow": 0,
    "thruStandBy": 0,
    "totalCheckedIn": 0,
    "totalOnBoard": 0
}
],
"inventoryLegCrossReferences": [
{
    "legKey": "NjM4NjQ4NLQ1NSMxNCB1NSNwIUpCITExMTEhICFTTEMhQk9TITMwMTM3NTM-",
    "number": 1,
    "marketInformation": {
        "identifier": {
            "identifier": "5600",
            "carrierCode": "NV",
            "opSuffix": null
        },
        "destination": "BOS",
        "origin": "SLC",
        "departureDate": "2025-08-07T04:47:17.0923185+00:00"
    }
}
],
"passengerCounts": {
    "checkedIn": 0,
    "clearedStandBy": 0,
    "clearedThruStandby": 0,
    "manifested": 1,
    "noShow": 1,
    "standBy": 0,
    "thruCheckedIn": 0,
    "thruManifested": 0,
    "thruNoShow": 0,
    "thruStandBy": 0,
    "totalCheckedIn": 0,
    "totalOnBoard": 0
},
"checkInLid": 810,
"codeShareIndicator": null,
"operatedByText": null,
"operatingIdentifier": null,
"legStatus": 0
}
]
}

```

Description: Example response for retrieving the manifest.

GET /api/nsk/v1/manifest/{legKey}/bags

Summary: Gets the manifest Passenger Bags for the specified leg. Sorted by recordLocator and then by Passenger. weightType 0 = Default, 1 = Pounds, 2 = Kilograms, (2 is current default for backwards compatibility)

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.
weightType	query	string	No	Weight type. 0 = Default, 1 = Pounds, 2 = Kilograms.
Enumeration values: 0 = Default, 1 = Pounds, 2 = Kilograms				

Responses

404 Response

200 Response

The manifest model.

Examples

Response 200

Manifest Bag Valid Response:

json



```
{
  "xxxxx - recordLocator": {
    "passenger1AlternateKey": [
      {
        "operationBagKey": "BagKey",
        "identifier": "NV1111",
        "baggageStatus": 1,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 0
      },
      {
        "operationBagKey": "BagKey2",
        "identifier": "NV1111",
        "baggageStatus": 3,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag2",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 1
      }
    ],
    "passenger2AlternateKey": [
      {
        "operationBagKey": "BagKey3",
        "identifier": "NV1111",
        "baggageStatus": 1,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag3",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 0
      }
    ]
  },
  "xxxxX2 - recordLocator 2": {
    "passenger3AlternateKey": [
      {
        "operationBagKey": "BagKey4",
        "identifier": "NV1111",
        "baggageStatus": 1,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag4",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 0
      }
    ],
    "passenger4AlternateKey": [
      {
        "operationBagKey": "BagKey5",
        "identifier": "NV1111",
        "baggageStatus": 1,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag5",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 0
      }
    ]
  }
}
```

```

        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 0
    },
    {
        "operationBagKey": "BagKey6",
        "identifier": "NV1111",
        "baggageStatus": 3,
        "nonStandard": true,
        "checkInDate": "2025-08-07T00:00:00+00:00",
        "osTag": "bagOsTag6",
        "compartment": 1,
        "taggedToStation": "SLC",
        "taggedToStationCity": "SLC",
        "weight": 10,
        "weightType": 1
    }
]
}
}

```

Description: This is a valid Manifest Bag Expected Response.

GET /api/nsk/v1/manifest/{legKey}/bookings

Summary: Gets the manifest bookings data for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.

Responses

404 Response

200 Response

The manifest model.

GET /api/nsk/v1/manifest/{legKey}/governmentSecurityInfo

Summary: Gets the manifest Passenger Government Security Information for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.

Responses

404 Response

200 Response

The manifest model by Booking then Passenger Information.

Examples

Response 200

Manifest Government Security Information Response:

json



```
{
  "data": {
    "xxxxxx - recordLocator": [
      {
        "userKey": "userKey",
        "personKey": "PersonKey",
        "passengerAlternateKey": "PassAltKey",
        "passengerKey": "PassKey",
        "checkInType": 0,
        "infant": {
          "nationality": "US",
          "dateOfBirth": "2025-08-07T04:47:17.1144714+00:00",
          "residentCountry": "US",
          "gender": 2,
          "name": {
            "first": "infFirst",
            "middle": "infMiddle",
            "last": "infLast",
            "title": "",
            "suffix": ""
          }
        },
        "name": {
          "first": "first",
          "middle": "Middle",
          "last": "Last",
          "title": "",
          "suffix": ""
        },
        "recordLocator": "ZZZZZZ - RecordLocator",
        "bookingStatus": 2,
        "passengerInformation": {
          "nationality": "US",
          "residentCountry": "US",
          "gender": 2,
          "dateOfBirth": "2025-08-07T04:47:17.1144833+00:00",
          "familyNumber": 1
        },
        "type": "ADT",
        "bookingDate": "2025-08-07T04:47:17.1144684+00:00",
        "isMissingVerifiedDocAdult": true,
        "isMissingVerifiedDocInfant": true,
        "passengerAddresses": [
          {
            "passengerAlternateKey": "PassAltKey",
            "countryName": "US",
            "lineOne": "Address",
            "companyName": "",
            "lineTwo": "",
            "phone": "8885551234",
            "lineThree": "",
            "stationCode": "SLC",
            "countryCode": "US",
            "emailAddress": "Yes@Yes.com",
            "provinceState": "UT",
            "status": 1,
            "city": "SLC",
            "passengerAddressKey": "PassKey",
            "postalCode": "88888"
          }
        ],
        "passengerTravelDocuments": [
          {
            "passengerTravelDocumentKey": "PassTravDocKey",
            "documentTypeCode": "P",
            "birthCountry": "US",
            "issuedByCode": "US",
            "name": {
              "first": "First",
              "middle": "Middle",
              "last": "Last",
              "title": "",
              "suffix": ""
            }
          }
        ]
      }
    ]
  }
}
```

```

        "title": "",
        "suffix": ""
    },
    "nationality": "US",
    "expirationDate": "2025-08-08T04:47:17.1144862+00:00",
    "number": "123456789",
    "issuedDate": "2025-08-07T04:47:17.1144868+00:00",
    "gender": 2,
    "dateOfBirth": "2025-08-07T04:47:17.1144854+00:00",
    "declaredGender": "Female",
    "placeOfBirth": "US",
    "placeOfIssue": "US"
},
{
    "passengerTravelDocumentKey": "InfPassTravDocKey",
    "documentTypeCode": "P",
    "birthCountry": "US",
    "issuedByCode": "US",
    "name": {
        "first": "infFirst",
        "middle": "infMiddle",
        "last": "infLast",
        "title": "",
        "suffix": ""
    },
    "nationality": "US",
    "expirationDate": "2025-08-08T04:47:17.1144878+00:00",
    "number": "123456789I",
    "issuedDate": "2025-08-07T04:47:17.114488+00:00",
    "gender": 2,
    "dateOfBirth": "2025-08-07T04:47:17.1144877+00:00",
    "declaredGender": "Female",
    "placeOfBirth": "US",
    "placeOfIssue": "US"
}
],
"passengerAddressRequirement": true,
"profileStatus": 0,
"authorizationList": null
}
]
}
}

```

Description: Example response for retrieving the manifest government security information.

GET /api/nsk/v2/manifest/{legKey}/passengerDetails

Summary: Gets the passenger details for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

401 Response

Authorization not allowed. Endpoint requires agent permissions.

403 Response

Role permissions are not met.

Examples

Response 200

Manifest with Passenger Details:

json



```
{
  "data": {
    "passengers": [
      {
        "passengerAlternateKey": "MjY00TQ3",
        "status": 2,
        "recordLocator": "ASDFGH",
        "createdDate": "2025-08-07T04:47:17.1406625Z",
        "name": {
          "first": "Justine",
          "middle": null,
          "last": "Doe",
          "title": null,
          "suffix": null
        },
        "profileStatus": 0,
        "infantTicketNumber": null,
        "weightCategory": 1,
        "ticketNumber": null,
        "origin": "SLC",
        "ticketIndicator": null,
        "destination": "BOS",
        "ssrCodes": null,
        "tripType": 1,
        "segmentBookingStatus": "HK",
        "connectingFlight": null,
        "connectingStation": null,
        "finalDestination": "BOS",
        "paymentMethods": [
          "CA"
        ],
        "comments": null,
        "infant": {
          "nationality": null,
          "dateOfBirth": "2025-07-07T00:00:00+00:00",
          "residentCountry": null,
          "gender": 1,
          "name": {
            "first": "Baby",
            "middle": null,
            "last": "Doe",
            "title": null,
            "suffix": null
          }
        },
        "customerProgram": null,
        "verifiedReference": null,
        "guestValueLevel": null,
        "paidStatus": 1
      },
      {
        "passengerAlternateKey": "MjY00TQ3",
        "status": 2,
        "recordLocator": "ASDFGH",
        "createdDate": "2025-08-07T04:47:17.1406727Z",
        "name": {
          "first": "Justine",
          "middle": null,
          "last": "Doe",
          "title": null,
          "suffix": null
        },
        "profileStatus": 0,
        "infantTicketNumber": null,
        "weightCategory": 3,
        "ticketNumber": null,
        "origin": "SLC",
        "ticketIndicator": null,
        "destination": "BOS",
        "ssrCodes": null,
        "tripType": 1
      }
    ]
  }
}
```

```

    "segmentBookingStatus": "HK",
    "connectingFlight": null,
    "connectingStation": null,
    "finalDestination": "BOS",
    "paymentMethods": [
        "CA"
    ],
    "comments": null,
    "infant": null,
    "customerProgram": null,
    "verifiedReference": null,
    "guestValueLevel": null,
    "paidStatus": 1
},
{
    "passengerAlternateKey": "MjY00TQ3",
    "status": 1,
    "recordLocator": "QWERTY",
    "createdDate": "2025-08-07T04:47:17.1406731Z",
    "name": {
        "first": "Justine",
        "middle": null,
        "last": "Doe",
        "title": null,
        "suffix": null
    },
    "profileStatus": 0,
    "infantTicketNumber": null,
    "weightCategory": 2,
    "ticketNumber": null,
    "origin": "SLC",
    "ticketIndicator": null,
    "destination": "BOS",
    "ssrCodes": null,
    "tripType": 1,
    "segmentBookingStatus": "HK",
    "connectingFlight": null,
    "connectingStation": null,
    "finalDestination": "BOS",
    "paymentMethods": [
        "CA"
    ],
    "comments": null,
    "infant": null,
    "customerProgram": null,
    "verifiedReference": null,
    "guestValueLevel": null,
    "paidStatus": 1
}
],
"summary": {
    "totalConfirmedMales": 1,
    "totalConfirmedFemales": 0,
    "totalConfirmedChildren": 1,
    "totalConfirmedInfants": 1,
    "totalWaitListConfirmedMales": 0,
    "totalWaitListConfirmedFemales": 0,
    "totalWaitListConfirmedChildren": 0,
    "totalWaitListConfirmedInfants": 0,
    "totalUnconfirmedMales": 0,
    "totalUnconfirmedFemales": 1,
    "totalUnconfirmedChildren": 0,
    "totalUnconfirmedInfants": 0
}
}
}
}

```

Description: Successful retrieval of manifest with passenger details.

Response 400

Invalid leg key:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'legKey' with value 'MjM4Nzc5NjQ0MDAwMDAwIUSWITU2MDAhICFTTEMhQk9TITQ20DAbc4+' is invalid.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'legKey' with value 'MjM4Nzc5NjQ0MDAwMDAwIUSWITU2MDAhICFTTEMhQk9TITQ20DAbc4+' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  ]
}

```

Description: The provided leg key is invalid and cannot be decoded correctly.

Response 403

Not allowed to retrieve manifest:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Authorization",
      "message": "Agent not authorized to view the Manifest report",
      "type": "Error",
      "details": null,
      "rawMessage": "Agent not authorized to view the Manifest report",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  ]
}

```

Description: The currently logged-in user is not authorized to retrieve the manifest with passenger details. Kindly enable the Roles > Departure Control > Manifest with connection Information and Roles > Departure Control > Selectees by Flight in the Management Console to successfully retrieve the manifest.

GET /api/dcs/v1/manifest/{legKey}/passengerDetails/weightBalance

Summary: Gets the weight balance passenger details for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

404 Response

200 Response

The passenger weight balance summary.

GET /api/nsk/v1/manifest/{legKey}/passengerSeatAssignments

Summary: Gets a list of seat assignment reports for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

404 Response

200 Response

The list of seat assignment reports.

GET /api/nsk/v1/manifest/{legKey}/segments

Summary: Gets the manifest segment data for the specified leg.

Description

This structure of manifest data most closely resembles the original v2 manifest structure without the duplicate data and performance issues.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.
includeNonOriginData	query	boolean	No	Include segments associated outside the requested leg.

Responses

404 Response

200 Response

The manifest model.

Examples

Response 200

Manifest By Segment Response:

json



```
{
  "data": {
    "segmentKey": null,
    "market": {
      "identifier": {
        "identifier": "5600",
        "carrierCode": "NV",
        "opSuffix": null
      },
      "destination": "BOS",
      "origin": "SLC",
      "departureDate": "2025-08-07T04:47:17.1088689+00:00"
    },
    "externalIdentifier": {
      "identifier": "5600",
      "carrierCode": "NV",
      "opSuffix": null
    },
    "legs": [
      {
        "legNumber": 1,
        "market": {
          "identifier": {
            "identifier": "5600",
            "carrierCode": "NV",
            "opSuffix": null
          },
          "destination": "BOS",
          "origin": "SLC",
          "departureDate": "2025-08-07T04:47:17.1088727+00:00"
        },
        "liftStatus": 2,
        "manifestOrigin": true,
        "manifestLegKey": "NjM4ODIzNDU2MDAwMDAwMDAwIUSWITU2MjUhICFERU4hU0xD"
      }
    ],
    "passengers": [
      {
        "passengerAlternateKey": "NDN2Lkb1",
        "baggageAllowanceUsed": false,
        "baggageAllowanceWeight": 0,
        "baggageAllowanceWeightType": 0,
        "cabinOfService": "Y",
        "changeReasonCode": null,
        "classOfService": "O",
        "fareBasisCode": "OFDAPI",
        "productClassCode": "FX",
        "seatProperties": [
          {
            "preferenceStatus": "Y",
            "propertyCode": "TRUE",
            "propertyTypeCode": "AISLE"
          }
        ],
        "ticketIndicator": null,
        "ticketNumber": null,
        "upgradeIndicator": false,
        "verifiedTravelDocs": null,
        "legs": [
          {
            "boardingSequence": 1,
            "boardingZone": 99,
            "compartmentDesignator": "F",
            "priorityCode": null,
            "priorityDate": "2025-08-07T04:47:17.1088999+00:00",
            "promptToVerifyId": false,
            "unitAssignmentWeight": 0,
            "unitDesignator": "2B"
          }
        ],
        "passengerSegmentOrderNumber": 0
      }
    ]
  }
}
```

```

        }
    ],
    "manifestSegmentKey": "NjM4ODIzNDU2MDAwMDAwMDAwIUsWITU2MjUhICFERU4hU0xD"
}
}

```

Description: Example response for retrieving the manifest by segment.

GET /api/nsk/v1/manifest/{legKey}/ssrs

Summary: Gets the manifest SSRs for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The manifest leg key.
includeNonOriginSsrs	query	boolean	No	Include SSRs associated to other legs or segments from the requested leg.

Responses

404 Response

200 Response

The manifest model.

Messages Endpoints

POST /api/nsk/v1/messages

Summary: Adds a new message item.

Description

Request Details

Request Body

The message queue.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The message queue.

400 Response

DELETE /api/nsk/v1/messages

Summary: Deletes a collection of message items.

Description

This endpoint does not perform additional verification that the messages were deleted. As long as there are no errors or exceptions, the delete operation is considered successful. A separate call to the GET /api/nsk/v1/messages endpoint can be made to verify that all the specified messages do not exist anymore.

Request Details

Request Body

The collection of message keys to delete.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/nsk/v2/messages

Summary: Retrieves a collection of teletype messages based on search criteria.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
LastPageKey	query	string	No	The key of the last item returned from paging.

`plaintext` 

|

| MessageQueueFilter | query | string | No | The filter that specifies which records to retrieve.

Enumeration values: 0 = NonHiddenRecords, 1 = HiddenRecords, 2 = All || MessageTypeCode | query | string | No | The message type code to search for. || SearchStartDate | query | string (date-time) | No | The start of created date range to search for. || SearchEndDate | query | string (date-time) | No | The end of created date range to search for. || PageSize | query | integer | No | The number of records to return when paging. || SearchValue | query | string | No | The search query. || SearchType | query | string | No | The type of search to be performed.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch |

Responses

200 Response

The message items.

404 Response

GET /api/nsk/v1/messages/{messageKey}

Summary: Retrieves a specific message item.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
messageKey	path	string	Yes	The key of the message to be retrieved.

Responses

200 Response

The message item.

404 Response

DELETE /api/nsk/v1/messages/{messageKey}

Summary: Deletes a message item.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
messageKey	path	string	Yes	The key of the message to delete.

Responses

200 Response

POST /api/nsk/v1/messages/teletype

Summary: Adds a new generic teletype message item.

Description

Request Details

Request Body

The generic teletype message request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The message queue.

400 Response

Onetimetravelnotifications Endpoints

POST /api/nsk/v2/oneTimeTravelNotifications

Summary: Creates a one time travel notification.

Description

Request Details

Request Body

The one time travel notification create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

500 Response

Examples

Request

Create One Time Travel Notification:

```
json   
{  
    "events": [  
        {  
            "type": 0  
        }  
    ],  
    "cultureCode": "en-US",  
    "timedEvents": [  
        {  
            "minutesToEvent": 10,  
            "type": 0  
        }  
    ],  
    "notificationDestination": {  
        "deliveryMethodCode": 0,  
        "destination": "example@gmail.com"  
    },  
    "marketInformation": {  
        "identifier": {  
            "identifier": "5611",  
            "carrierCode": "NV",  
            "opSuffix": null  
        },  
        "destination": "BOS",  
        "origin": "SLC",  
        "departureDate": "2025-08-07T04:47:15.97742Z"  
    },  
    "deviceName": null  
}
```

Description: Example request when creating a one time travel notification.

Create One Time Travel Notification With Events Only:

```
json   
{  
    "events": [  
        {  
            "type": 0  
        }  
    ],  
    "cultureCode": "en-US",  
    "timedEvents": null,  
    "notificationDestination": {  
        "deliveryMethodCode": 0,  
        "destination": "example@gmail.com"  
    },  
    "marketInformation": {  
        "identifier": {  
            "identifier": "5611",  
            "carrierCode": "NV",  
            "opSuffix": null  
        },  
        "destination": "BOS",  
        "origin": "SLC",  
        "departureDate": "2025-08-07T04:47:15.977421Z"  
    },  
    "deviceName": null  
}
```

Description: Example request when creating a one time travel notification with events only.

Create One Time Travel Notification With Timed Events Only:

```
json

{
  "events": null,
  "cultureCode": "en-US",
  "timedEvents": [
    {
      "minutesToEvent": 10,
      "type": 0
    }
  ],
  "notificationDestination": {
    "deliveryMethodCode": 0,
    "destination": "example@gmail.com"
  },
  "marketInformation": {
    "identifier": {
      "identifier": "5611",
      "carrierCode": "NV",
      "opSuffix": null
    },
    "destination": "BOS",
    "origin": "SLC",
    "departureDate": "2025-08-07T04:47:15.9774224Z"
  },
  "deviceName": null
}
```

Description: Example request when creating a one time travel notification with timed events only.

Response 400

Invalid E-mail Address:

```
json

{
  "errors": [
    {
      "id": "63dd49c7-59f7-5a0c-d07a-d249785314e9",
      "code": "nsk-server:InvalidEmailAddress",
      "message": "nsk-server:InvalidEmailAddress",
      "type": "Validation",
      "details": {
        "message": "@gmail.com is not a valid email address."
      },
      "rawMessage": "@gmail.com is not a valid email address.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when e-mail address is invalid

Invalid Phone Number:

```
json
{
  "errors": [
    {
      "id": "63dd49c7-59f7-5a0c-d07a-d249785314e9",
      "code": "nsk-server:InvalidPhoneNumber",
      "message": "nsk-server:InvalidPhoneNumber",
      "type": "Validation",
      "details": {
        "message": "ABD is not a valid email phone number."
      },
      "rawMessage": "ABD is not a valid email phone number.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when phone number is invalid

Null Notification Destination:

```
json
{
  "errors": [
    {
      "id": "ca448f1f-9dae-afa8-bc79-423a6aaa9c56",
      "code": "nsk:TravelNotification:NullDestination",
      "message": "nsk:TravelNotification:NullDestination",
      "type": "Validation",
      "details": null,
      "rawMessage": "The destination of the subscription should not be blank.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with a null or empty notification destination.

Invalid Device Name for Push Notification:

```
json
{
  "errors": [
    {
      "id": "6ac6116b-85f8-f47d-f100-99338b6f8287",
      "code": "nsk:DeviceName:InvalidDeviceName",
      "message": "nsk:DeviceName:InvalidDeviceName",
      "type": "Validation",
      "details": null,
      "rawMessage": "The device name should be set for notification of type Push",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an invalid device name for push notification

Invalid Device Name for Email and Phone Notification:

```
json
{
  "errors": [
    {
      "id": "6ac6116b-85f8-f47d-f100-99338b6f8287",
      "code": "nsk:DeviceName:InvalidDeviceName",
      "message": "nsk:DeviceName:InvalidDeviceName",
      "type": "Validation",
      "details": null,
      "rawMessage": "The device name should not be set for notification of type Email",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present"
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with a device name that should not be set for email and phone notifications

Empty Event Type:

```
json
{
  "errors": [
    {
      "id": "135f4684-f240-3988-e042-69aad6e5cce4",
      "code": "OneTimeNotificationEventCreateRequest.Type:RequiredAttribute",
      "message": "OneTimeNotificationEventCreateRequest.Type:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "OneTimeNotificationEventCreateRequest",
        "member": "Type"
      },
      "rawMessage": "The OneTimeNotificationEventCreateRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present"
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an empty event type

Empty Timed Event Type:

```
json
{
  "errors": [
    {
      "id": "6214d832-864c-a521-1e31-c24190043349",
      "code": "NotificationTimedEventCreateRequest.Type:RequiredAttribute",
      "message": "NotificationTimedEventCreateRequest.Type:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "NotificationTimedEventCreateRequest",
        "member": "Type"
      },
      "rawMessage": "The NotificationTimedEventCreateRequest field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present"
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an empty timed event type

Invalid Minutes to Event:

```
json

{
  "errors": [
    {
      "id": "d9f64797-ec60-93fb-dfe4-4bf4c0c94dd5",
      "code": "nsk:MinutesToEvent:InvalidMinutesToEventValue",
      "message": "nsk:MinutesToEvent:InvalidMinutesToEventValue",
      "type": "Validation",
      "details": {
        "minutesToEvent": "5"
      },
      "rawMessage": "The minutes to event value of '5' is invalid for timed notification even type 'Arrival'.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an invalid minutes to event value

Invalid Event Type:

```
json

{
  "errors": [
    {
      "id": "d48df503-4c4d-30a8-0862-a3e03b0cbb1f",
      "code": "nsk:NotificationEvent:InvalidEventType",
      "message": "nsk:NotificationEvent:InvalidEventType",
      "type": "Validation",
      "details": {
        "eventType": "2"
      },
      "rawMessage": "The event type '2' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an invalid event type

Invalid Timed Event Type:

```
json

{
  "errors": [
    {
      "id": "b1c5c484-0f33-f847-82d8-28aedb3975c2",
      "code": "nsk:NotificationEvent:InvalidTimedEventType",
      "message": "nsk:NotificationEvent:InvalidTimedEventType",
      "type": "Validation",
      "details": {
        "timedEventType": "2"
      },
      "rawMessage": "The timed event type '2' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when creating a one time travel notification with an invalid timed event type

Empty Subscription Events List:

```
json

{
  "errors": [
    {
      "id": "7e18d08c-bbbc-2d2f-6c2b-9456d84670ac",
      "code": "nsk:NotificationEvent:NullEvents",
      "message": "nsk:NotificationEvent:NullEvents",
      "type": "Validation",
      "details": null,
      "rawMessage": "The list of subscription events should not be empty.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the list of subscription events are empty.

Invalid Culture Code:

```
json

{
  "errors": [
    {
      "id": "20c566a3-e928-1afb-edd6-b9725789060b",
      "code": "nsk:TravelNotification:InvalidCultureCode",
      "message": "nsk:TravelNotification:InvalidCultureCode",
      "type": "Validation",
      "details": {
        "cultureCode": "dn-US"
      },
      "rawMessage": "The culture code dn-US is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the culture code is invalid.

Response 500

Existing Notification:

```
json

{
  "errors": [
    {
      "id": "70c46bde-ece4-508c-ce33-c59f652038ae",
      "code": "nsk-server:SubscriptionCommitFailed",
      "message": "nsk-server:SubscriptionCommitFailed",
      "type": "Error",
      "details": null,
      "rawMessage": "Request already received for 20240408 NV5611 SLCBOS with email1@test.com; email2@test.com. If an update is required, use PUT /v1/travel-notifications/{id} instead.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when a one time travel notification already exists when creating with the same destination address.

Invalid Market Information:

```
json
{
  "errors": [
    {
      "id": "a0c9c3ac-f0a0-a1de-cb55-2696fc0c8562",
      "code": "nsk-server:WarningException",
      "message": "nsk-server:WarningException",
      "type": "Error",
      "details": null,
      "rawMessage": "The flight being subscribed to (20240415 NV 561 SLCBOS) cannot be found ",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Example response when the market information is invalid.

GET /api/nsk/v2/oneTimeTravelNotifications/{subscriptionNumber}

Summary: Gets a one time travel notification through a subscription number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
subscriptionNumber	path	string	Yes	The subscription number.
destination	query	string	No	The notification destination.

Responses

200 Response

400 Response

404 Response

Examples

Response 200

Retrieve One Time Travel Notification Response:

```

json
{
  "data": {
    "expirationDate": "2025-08-10T04:47:15.9848088Z",
    "cultureCode": "US",
    "subscriptionNumber": "446776225723",
    "notificationDestination": {
      "deliveryMethodCode": 0,
      "destination": "test@gmail.com"
    },
    "deviceName": null,
    "events": [
      {
        "createdDate": "2025-08-07T04:47:15.9848127Z",
        "type": 0
      }
    ],
    "timedEvents": [
      {
        "createdDate": "2025-08-07T04:47:15.9848156Z",
        "minutesToEvent": 10,
        "type": 0
      }
    ],
    "marketInformation": {
      "identifier": {
        "identifier": "5611",
        "carrierCode": "NV",
        "opSuffix": null
      },
      "destination": "SLC",
      "origin": "BOS",
      "departureDate": "2025-08-08T04:47:15.9848169Z"
    }
  }
}

```

Description: Example response for successfully retrieving a One Time Travel Notification by Subscription Number

DELETE /api/nsk/v2/oneTimeTravelNotifications/{subscriptionNumber}

Summary: Deletes a one time travel notification through a subscription number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
subscriptionNumber	path	string	Yes	The subscription number.
destination	query	string	No	The notification destination.

Responses

200 Response

400 Response

Examples

Response 400

Subscription number not found:

```
json

{
  "errors": [
    {
      "id": "04c4da13-bb82-1816-8a4a-c3ef561516e7",
      "code": "nsk:TravelNotification:SubscriptionNumberNotFound",
      "message": "nsk:TravelNotification:SubscriptionNumberNotFound",
      "type": "Validation",
      "details": null,
      "rawMessage": "The subscription number could not be found.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."}
  ],
  "data": null
}
```

Description: Example response when the subscription number is not found

Organizationgroup Endpoints

POST /api/nsk/v2/organizationGroup

Summary: Creates a new organization group and adds an organization to it.

Description

Request Details

Request Body

The organization group create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Request

Organization Group Create Request:

```
json

{
  "organizationGroupCode": "ABC",
  "organizationGroupName": "Group Name",
  "organization": {
    "organizationCode": "AAA",
    "name": "Organization Name"
  }
}
```

Description: Creates a new organization group

Organizationgroups Endpoints

PUT /api/nsk/v1/organizationGroups/{organizationGroupCode}

Summary: Modifies an organization group by updating the group name or adding an organization to the organization group.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationGroupCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

404 Response

200 Response

DELETE /api/nsk/v1/organizationGroups/{organizationGroupCode}

Summary: Deletes an organization group. This should not be used if the organization group needs to exist.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationGroupCode	path	string	Yes	No description

Responses

200 Response

PUT /api/nsk/v1/organizationGroups/{organizationGroupCode}/organizations

Summary: Updates an organization group by removing all organizations

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationGroupCode	path	string	Yes	No description

Responses

404 Response

200 Response

PUT /api/nsk/v1/organizationGroups/{organizationGroupCode}/organizations/{organizationCode}

Summary: Updates an organization group by removing an organization

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationGroupCode	path	string	Yes	No description
organizationCode	path	string	Yes	No description

Responses

404 Response

200 Response

Organizations Endpoints

GET /api/nsk/v1/organizations/groups

Summary: Retrieves the organization groups.

Description

Request Details

Responses

404 Response

200 Response

GET /api/nsk/v1/organizations/groups/{organizationGroupCode}

Summary: Retrieves the details of an organization group.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationGroupCode	path	string	Yes	No description

Responses

404 Response

200 Response

Organizations2 Endpoints

POST /api/nsk/v1/organizations2

Summary: Create a new organization.

Description

Requires agent permissions. The organization code is a generic code for all of the organization types. Each individual type has its own code and validation. If the organization status is not pending the organization code should follow these validation rules based on type:

plaintext



Third Party: 3 to 10 characters in length with alphanumeric values.
Travel Agency: 7 to 10 characters in length with numeric values.
Carrier: 2 to 3 characters in length with alphanumeric values.

If no organization code is provided then one will be generated and returned on the header.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The organization created.

GET /api/nsk/v2/organizations2

Summary: Searches for organizations.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
OrganizationCode	query	string	No	The organization code.

plaintext



|

| Type | query | string | No | Organization type.

Enumeration values: 0 = Default, 1 = Master, 2 = Carrier, 3 = TravelAgency, 4 = ThirdParty || Status | query | string | No | The organization status.

Enumeration values: 0 = Default, 1 = Active, 2 = Cancelled, 3 = Pending || ParentOrganizationCode | query | string | No | The parent organization code. || CompanyName | query | string | No | The organization company name. || City | query | string | No | The organization company city. || PostalCode | query | string | No | The organization company postal code. || PagedItemIndex | query | integer (int64) | No | Used to specify the index for skipping values when paging. || PageSize | query | integer | No | Used to specify the maximum results to return. || MatchCriteria.OrganizationCode | query | string | No | The type of string search for organization code.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || MatchCriteria.ParentOrganizationCode | query | string | No | The type of string search for parent organization code.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || MatchCriteria.CompanyName | query | string | No | The type of string search for organization company name.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || MatchCriteria.City | query | string | No | The type of string search for organization company city.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || MatchCriteria.PostalCode | query | string | No | The type of string search for organization company postal code.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch |

Responses

404 Response

200 Response

The organization records.

GET /api/nsk/v1/organizations2/{organizationCode}

Summary: Retrieves the organization.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Responses

404 Response

200 Response

The organization.

PUT /api/nsk/v1/organizations2/{organizationCode}

Summary: Updates an organization.

Description

Requires agent permissions. The organization code is a generic code for all of the organization types. Each individual type has its own code and validation. If the organization status is not pending the organization code should follow these validation rules based on type:

plaintext	
<p>Third Party: 3 to 10 characters in length with alphanumeric values. Travel Agency: 7 to 10 characters in length with numeric values. Carrier: 2 to 3 characters in length with alphanumeric values.</p>	

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The organization created.

PATCH /api/nsk/v1/organizations2/{organizationCode}

Summary: Updates an organization.

Description

Requires agent permissions. The organization code is a generic code for all of the organization types. Each individual type has its own code and validation. If the organization status is not pending the organization code should follow these validation rules based on type:

plaintext



Third Party: 3 to 10 characters in length with alphanumeric values.
Travel Agency: 7 to 10 characters in length with numeric values.
Carrier: 2 to 3 characters in length with alphanumeric values.

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The organization created.

GET /api/nsk/v1/organizations2/{organizationCode}/account

Summary: Retrieves the organization account and collections based on the organization code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.

Responses

404 Response

200 Response

POST /api/nsk/v1/organizations2/{organizationCode}/account

Summary: Creates an organization account based on the organization code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.

Request Body

The create organization account request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST /api/nsk/v1/organizations2/{organizationCode}/account/childAccountTransactions

Summary: Adjust the available amount of a child account.

Description

The known list of parent-child relations are as follows:

- Dependent accounts are not supported.
- A parent's account of type credit may have children of type supplemental or credit.
- A parent's account of type prepaid may have children of type prepaid.
- No other combination of parent-child relations are supported.

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The account reference number for the child organization.

Request Body

The child account transaction request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

PUT /api/nsk/v1/organizations2/{organizationCode}/account/childAccountTransactions

Summary: Adjust the credit limit of a child account.

Description

The known list of parent-child relations are as follows:

- Dependent accounts are not supported.
- A parent's account of type credit may have children of type supplemental or credit.
- A parent's account of type prepaid may have children of type prepaid.
- No other combination of parent-child relations are supported.

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The account reference number for the child organization.

Request Body

The child account transaction request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

PUT /api/nsk/v1/organizations2/{organizationCode}/account/status

Summary: Updates the organization account status.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.
status	query	string	No	The allowed account status.
Enumeration values: 0 = Open, 1 = Closed, 2 = AgencyInactive, 3 = Unknown				

Responses

200 Response

500 Response

POST /api/nsk/v1/organizations2/{organizationCode}/account/transactions

Summary: Creates account transactions based on data in the request and the organization code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.

Request Body

The transaction request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

GET /api/nsk/v2/organizations2/{organizationCode}/account/transactions

Summary: Retrieves the organization account transactions based on the organization code and data in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

```
plaintext
```

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json
{
  "errors": [
    {
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "TransactionSearchRequestv2",
        "member": "StartDate"
      },
      "rawMessage": "The TransactionSearchRequestv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```
json
{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when end date is before start date.

POST /api/nsk/v1/organizations2/{organizationCode}/commissionRates

Summary: Create a commission rate.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/organizations2/{organizationCode}/commissionRates/{commissionRateCode}

Summary: Get a commission rate.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
commissionRateCode	path	string	Yes	No description

Responses

200 Response

DELETE

/api/nsk/v1/organizations2/{organizationCode}/commissionRates/{commissionRateCode}

Summary: Delete a commission rate.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
commissionRateCode	path	string	Yes	No description

Responses

201 Response

POST /api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers

Summary: Create a company phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT

/api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers/{phoneNumberType}

Summary: Updates a company phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
phoneNumberType	path	string	Yes	
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

DELETE

/api/nsk/v1/organizations2/{organizationCode}/company/phoneNumbers/{phoneNumberType}

Summary: Deletes a company phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
phoneNumberType	path	string	Yes	
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Responses

200 Response

POST /api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers

Summary: Create a contact phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT

[`/api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers/{phoneNumberType}`](#)

Summary: Updates a contact phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
phoneNumberType	path	string	Yes	
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

DELETE

[`/api/nsk/v1/organizations2/{organizationCode}/contact/phoneNumbers/{phoneNumberType}`](#)

Summary: Deletes a contact phone number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
phoneNumberType	path	string	Yes	
Enumeration values: 0 = Other, 1 = Home, 2 = Work, 3 = Mobile, 4 = Fax				

Responses

200 Response

[`POST /api/nsk/v1/organizations2/{organizationCode}/externalAccounts`](#)

Summary: Create an external account.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/organizations2/{organizationCode}/externalAccounts/{externalAccountKey}

Summary: Get an external account.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
externalAccountKey	path	string	Yes	No description

Responses

200 Response

DELETE /api/nsk/v1/organizations2/{organizationCode}/externalAccounts/{externalAccountKey}

Summary: Delete an external account.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	No description
externalAccountKey	path	string	Yes	No description

Responses

201 Response

GET /api/nsk/v1/organizations2/{organizationCode}/processSchedules

Summary: Returns a list of Process Schedules for the organization.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.

Responses

200 Response
400 Response
404 Response

POST /api/nsk/v1/organizations2/{organizationCode}/processSchedules/{processScheduleId}

Summary: Adds a process schedule to the organization.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.
processScheduleId	path	integer (int64)	Yes	The ID of the process schedule.

Responses

201 Response

DELETE /api/nsk/v1/organizations2/{organizationCode}/processSchedules/{processScheduleId}

Summary: Deletes a process schedule from the organization.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
organizationCode	path	string	Yes	The organization code.
processScheduleId	path	integer (int64)	Yes	The ID of the process schedule.

Responses

200 Response

GET /api/nsk/v1/organizations2/allotments

Summary: Retrieves all allotments for the specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	query	string	No	The leg key.

Responses

200 Response

A list of allotments.

400 Response

Examples

Response 200

Allotment response:

```
json   
{  
    "allotmentBasisCode": "NEGO1",  
    "authorizedUnits": 10,  
    "allotted": 10,  
    "nest": 1,  
    "classOfService": "NEGO",  
    "type": 4,  
    "rank": 0,  
    "seatsSold": 5,  
    "latestAdvanceReservation": 0,  
    "status": 1,  
    "legNestKey": "MQ--",  
    "legClassKey": "Mjk1MTAwNCExIU5FR08-",  
    "legKey": null,  
    "marketInformation": null  
}
```

Description: Returns an allotment response

GET /api/nsk/v1/organizations2/allotments/{allotmentBasisCode}

Summary: Retrieves a single allotment for the specified leg using the given allotment basis code.

Description

The allotment basis code should be substituted with allotment class code in Fare Manager Plus environments.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	query	string	No	The leg key.
allotmentBasisCode	path	string	Yes	The allotment basis code.

Responses

200 Response

An allotment.

400 Response

Examples

Response 200

Allotment response:

```

json

{
  "allotmentBasisCode": "NEGO1",
  "authorizedUnits": 10,
  "allotted": 10,
  "nest": 1,
  "classOfService": "NEGO",
  "type": 4,
  "rank": 0,
  "seatsSold": 5,
  "latestAdvanceReservation": 0,
  "status": 1,
  "legNestKey": "MQ--",
  "legClassKey": "Mjk1MTAwNCExIU5FR08-",
  "legKey": null,
  "marketInformation": null
}

```

Description: Returns an allotment response

GET /api/nsk/v1/organizations2/allotments/byAllotmentBasisCode/{allotmentBasisCode}

Summary: Retrieves all allotments for the specified allotment basis code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
allotmentBasisCode	path	string	Yes	The allotment basis code.

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 200

Allotment response:

```

json

{
  "allotmentBasisCode": "NEGO1",
  "authorizedUnits": 10,
  "allotted": 10,
  "nest": 1,
  "classOfService": "NEGO",
  "type": 4,
  "rank": 0,
  "seatsSold": 5,
  "latestAdvanceReservation": 0,
  "status": 1,
  "legNestKey": "MQ--",
  "legClassKey": "Mjk1MTAwNCExIU5FR08-",
  "legKey": null,
  "marketInformation": null
}

```

Description: Returns an allotment response

POST

/api/nsk/v1/organizations2/contracts/basis/{allotmentBasisCode}/allotments/legs/{legKey}

Summary: Creates an allotment for the specified leg.

Description

Due to system limitations, if this endpoint is used the inventory cache will need to be refreshed NegoAllotment and ProrataAllotment are the only allotment class types available. These values align with the ClassType enum.

plaintext



```
The allotment basis code should be substituted with allotment class code in Fare Manager Plus environments.
```

Request Details

Parameters

Name	Location	Type	Required	Description
allotmentBasisCode	path	string	Yes	The allotment basis code
legKey	path	string	Yes	The leg key

Request Body

The allotment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Request

Allotment Request:

json



```
{
  "authorizedUnits": 10,
  "allotted": 10,
  "nest": 1,
  "classType": 4,
  "rank": 1,
  "latestAdvanceReservation": 0,
  "status": 1
}
```

Description: Creates a new allotment

Response 201

Allotment Created Response:

```
json
{
  "allotmentBasisCode": "ABC",
  "legKey": "LTg10DQ0NzA2NDg0NzMzOTYyNzYhQUEhMTIzNCEhU0xDIUJPUyExMw--"
}
```

Description: Creates a new allotment basis

PUT /api/nsk/v1/organizations2/contracts/basis/{allotmentBasisCode}/allotments/legs/{legKey}

Summary: Updates an allotment for the specified leg.

Description

Due to system limitations, if this endpoint is used the inventory cache will need to be refreshed. NegoAllotment and ProrataAllotment are the only allotment class types available. These values align with the ClassType enum. The allotment basis code should be substituted with allotment class code in Fare Manager Plus environments.

Request Details

Parameters

Name	Location	Type	Required	Description
allotmentBasisCode	path	string	Yes	The allotment basis code
legKey	path	string	Yes	The leg key

Request Body

The allotment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Request

Allotment Request:

```
json
{
  "authorizedUnits": 10,
  "allotted": 10,
  "nest": 1,
  "classType": 4,
  "rank": 1,
  "latestAdvanceReservation": 0,
  "status": 1
}
```

Description: Creates a new allotment

GET /api/nsk/v2/organizations2/processSchedule/{processScheduleId}

Summary: Returns a single Process Schedule object.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
processScheduleId	path	integer (int64)	Yes	The ID of the process schedule to retrieve.

Responses

200 Response

404 Response

Examples

Response 200

Successfully returned process schedule:

```
json   
{  
    "processScheduleId": 123,  
    "name": "Process Schedule A",  
    "beginDate": "2025-08-07T04:47:18.225Z",  
    "frequencyType": 0,  
    "daysOfWeek": [  
        1,  
        3,  
        5  
    ],  
    "frequency": 2,  
    "monthDay": 0,  
    "monthWeek": 0,  
    "inActive": false  
}
```

Description: Example response for successful retrieval of a process schedule. DaysOfweek is a list in which None=0, Monday=1, Tuesday=2, through Sunday=7.

GET /api/nsk/v2/organizations2/processSchedules

Summary: Returns a list of active process schedules.

Description

Request Details

Responses

200 Response

Examples

Response 200

Successfully returned list of all process schedules:

json

```
[  
  {  
    "processScheduleId": 123,  
    "name": "Process Schedule - Runs every Mon Wed Fri",  
    "beginDate": "2025-08-07T04:47:18.22419+00:00",  
    "frequencyType": 0,  
    "daysOfWeek": [  
      1,  
      3,  
      5  
    ],  
    "frequency": 1,  
    "monthDay": 0,  
    "monthWeek": 0,  
    "inActive": false  
  },  
  {  
    "processScheduleId": 124,  
    "name": "Process Schedule - Runs every third Tuesday",  
    "beginDate": "2025-08-07T04:47:18.2241953+00:00",  
    "frequencyType": 1,  
    "daysOfWeek": [  
      2  
    ],  
    "frequency": 3,  
    "monthDay": 0,  
    "monthWeek": 0,  
    "inActive": false  
  },  
  {  
    "processScheduleId": 125,  
    "name": "Process Schedule - Runs the second Saturday of each month",  
    "beginDate": "2025-08-07T04:47:18.2241956+00:00",  
    "frequencyType": 2,  
    "daysOfWeek": [  
      6  
    ],  
    "frequency": 1,  
    "monthDay": 0,  
    "monthWeek": 2,  
    "inActive": false  
  },  
  {  
    "processScheduleId": 126,  
    "name": "Process Schedule - Runs every day",  
    "beginDate": "2025-08-07T04:47:18.2241958+00:00",  
    "frequencyType": 0,  
    "daysOfWeek": [  
      0  
    ],  
    "frequency": 1,  
    "monthDay": 0,  
    "monthWeek": 0,  
    "inActive": false  
  },  
  {  
    "processScheduleId": 127,  
    "name": "Process Schedule - Runs every other day",  
    "beginDate": "2025-08-07T04:47:18.224196+00:00",  
    "frequencyType": 0,  
    "daysOfWeek": [  
      0  
    ],  
    "frequency": 2,  
    "monthDay": 0,  
    "monthWeek": 0,  
    "inActive": false  
  }  
]
```

Description: Example response for successful retrieval of all process schedules. DaysOfWeek is a list in which None=0, Monday=1, Tuesday=2, through Sunday=7.

POST /api/nsk/v1/organizations2/register

Summary: Requests to register a new organization.

Description

Note: This is only a request. An agent will need to approve the action.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Payments Endpoints

GET /api/nsk/v1/payments

Summary: Gets the payment data based on the specified bank identification number and date range.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
BankIdentificationNumber	query	integer (int32)	No	The bank identification number to search for. This is the first 6 digits of the card number.

plaintext 

| StartDate | query | string (date-time) | No | The start date to search for. This refers to the date the payment was made. || EndDate | query | string (date-time) | No | The end date. || PageSize | query | integer | No | The number of payment records per request. || LastPageKey | query | string | No | The key for the last payment record returned from paged response. |

Responses

200 Response

400 Response

Examples

Response 200

With matching payment data from card payments:

json



```
{
  "data": [
    {
      "lastPageKey": "0A--",
      "totalCount": 10,
      "paymentRecords": [
        {
          "mjc2NDIzIV1EVEQ3RQ--": [
            {
              "accountNumber": "XXXXXXXXXXXX1111",
              "paymentDate": "2025-08-07T04:47:15.9325749Z",
              "passengerName": {
                "first": "Jane",
                "middle": null,
                "last": "Doe",
                "title": null,
                "suffix": null
              },
              "departureDate": "2025-08-07T04:47:15.932577+00:00",
              "status": 3,
              "recordLocator": "XYZ456",
              "origin": "SLC",
              "destination": "BOS"
            },
            {
              "accountNumber": "XXXXXXXXXXXX1111",
              "paymentDate": "2025-08-07T04:47:15.9325797Z",
              "passengerName": {
                "first": "Jane",
                "middle": null,
                "last": "Doe",
                "title": null,
                "suffix": null
              },
              "departureDate": "2025-08-08T04:47:15.9325798+00:00",
              "status": 3,
              "recordLocator": "XYZ456",
              "origin": "BOS",
              "destination": "SLC"
            }
          ],
          "mjc2NDY1IVRFRT1URQ--": [
            {
              "accountNumber": "XXXXXXXXXXXX1111",
              "paymentDate": "2025-08-07T04:47:15.9325815Z",
              "passengerName": {
                "first": "John",
                "middle": null,
                "last": "Doe",
                "title": null,
                "suffix": null
              },
              "departureDate": "2025-08-07T04:47:15.9325816+00:00",
              "status": 4,
              "recordLocator": "ABC123",
              "origin": "SLC",
              "destination": "BOS"
            },
            {
              "accountNumber": "XXXXXXXXXXXX1111",
              "paymentDate": "2025-08-07T04:57:15.9325818Z",
              "passengerName": {
                "first": "John",
                "middle": null,
                "last": "Doe",
                "title": null,
                "suffix": null
              },
              "departureDate": "2025-08-07T04:47:15.9325823+00:00",
              "status": 3,
              "recordLocator": "ABC123",
              "origin": "SLC",
              "destination": "BOS"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        }
    ]
}
}
```

Description: Example response for successful retrieval of card payment data by BIN.

No matching payment data:

```
json

{
  "data": {
    "lastPageKey": "MTA-",
    "totalCount": 0,
    "paymentRecords": null
  }
}
```

Description: Example response when no payment data matches the query.

Response 400

Invalid BIN value:

```
json

{
  "errors": [
    {
      "code": "FindPaymentsByBinRequest.BankIdentificationNumber:RequiredAttribute",
      "message": "The FindPaymentsByBinRequest field is required.",
      "type": "Validation",
      "number": null,
      "details": {
        "member": "BankIdentificationNumber",
        "model": "FindPaymentsByBinRequest",
        "validation": "RangeAttribute"
      }
    }
  ]
}
```

Description: Example error if the BIN value is not specified or invalid

Out of range BIN value:

```
json

{
  "errors": [
    {
      "code": "FindPaymentsByBinRequest.BankIdentificationNumber:RangeAttribute",
      "message": "The field FindPaymentsByBinRequest must be between 100000 and 999999.",
      "type": "Validation",
      "number": null,
      "details": {
        "member": "BankIdentificationNumber",
        "model": "FindPaymentsByBinRequest",
        "validation": "RangeAttribute"
      }
    }
  ]
}
```

Description: Example error if the BIN value is out of range.

Start date not specified:

```
json
```

```
{  
  "errors": [  
    {  
      "code": "nsk:Validation:StartDateRequired",  
      "message": "The start date is required for the request.",  
      "type": "Validation",  
      "number": null,  
      "details": {  
        "member": "StartDate",  
        "model": "FindPaymentsByBinRequest",  
        "validation": "RequiredAttribute"  
      }  
    }  
  ]  
}
```

Description: Example error if the start date is not specified.

End date before start date:

```
json
```

```
{  
  "errors": [  
    {  
      "code": "nsk:Validation:EndDateBeforeBeginDate",  
      "message": "The requested end date of 1/1/2024 occurs before the requested begin date of 1/30/2024.",  
      "type": "Validation",  
      "number": null,  
      "details": null  
    }  
  ]  
}
```

Description: Example error if the end date is before the start date.

Agent not allowed:

```
json
```

```
{  
  "errors": [  
    {  
      "code": "nsk:Payment:SearchPaymentsByBinNotAllowed",  
      "message": "Searching payment records by BIN is not allowed based on the current user's permissions.",  
      "type": "Validation",  
      "number": null,  
      "details": null  
    }  
  ]  
}
```

Description: Example error if the agent is not authorized to search payments by bank identification number.

Do Not Persist External Payment Details Enabled:

```

json
{
  "errors": [
    {
      "code": "nsk:SystemSettings:DoNotPersistExternalPaymentDetailsEnabled",
      "message": "The Do not Persist External Payment Details premium feature is enabled.",
      "type": "Validation",
      "number": null,
      "details": null
    }
  ]
}

```

Description: Example error if The Do not Persist External Payment Details premium feature is enabled.

Persons Endpoints

POST /api/nsk/v1/persons

Summary: Creates a new person record (used by agents).

Description

This endpoint requires agent permissions.

Request Details

Request Body

The person create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Location (via header) where to get created person.

GET /api/nsk/v2/persons

Summary: Search person records.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
LastIndex	query	integer (int32)	No	The index of the last paged item. This can be found on the response from the previous set of paged results.

```
plaintext
```

| Username | query | string | No | The person user name to search by. || FirstName | query | string | No | The person's first name to search by. || LastName | query | string | No | The person's last name to search by. || FirstNameMatching | query | string | No | The matching criteria for the person's first name.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || CustomerNumber | query | string | No | The person's customer number. || PhoneNumber | query | string | No | The phone number to search by. || EmailAddress | query | string | No | The email address to search by. || ProgramNumber | query | string | No | The program number to search by. || ProgramCode | query | string | No | The program code to search by. || Type | query | string | No | The person type to search by.

Enumeration values: 0 = None, 1 = Customer, 2 = Agent || NationalIdNumber | query | string | No | The national ID number to search by. || NationalIdNumberMatching | query | string | No | The matching criteria for the national ID number.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || ReturnCount | query | integer | No | The number of results to return. || ActiveOnly | query | boolean | No | Include active person records only. |

Responses

404 Response

200 Response

The collection of person records.

GET /api/nsk/v1/persons/{personKey}

Summary: Retrieves a specific person by key.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The person.

PUT /api/nsk/v1/persons/{personKey}

Summary: Updates the person record basic information.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The person edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}

Summary: Patches the person records basic information.

Description

- Only need to send in the data that is being requested to be updated.
- This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The person patch request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}

Summary: Deletes a person by setting the record to terminated.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/account

Summary: Retrieves the person account and credits based on the person key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.

Responses

404 Response

200 Response

POST /api/nsk/v1/persons/{personKey}/account

Summary: Creates the person account based on the person key and data in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.

Request Body

The create account request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST /api/nsk/v1/persons/{personKey}/account/collection

Summary: Creates an account collection based on the person key and data in the request.

Description

This endpoint will create a new account collection and a transaction if no account transaction exists or no matching collection is found. An account collection is unique based on the transaction code and the expiration date. If a matching account collection is found, a transaction for that collection is generated and the account collection will be updated. See /api/nsk/v1/resources/accountTransactionCodes for a list of available account transaction codes.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.

Request Body

The create account collection request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

POST /api/nsk/v1/persons/{personKey}/account/collection/{accountCollectionKey}/transactions

Summary: Creates an account collection transaction based on the person key, the account collection key, and data in the request.

Description

This endpoint will add a transaction to an existing account collection. If there are no account collections, or the account collection key does not match an existing collection, the request will fail. To add a new account collection, see /api/nsk/v1/persons/{personKey}/account/collection.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
accountCollectionKey	path	string	Yes	The account collection key.

Request Body

The create transaction request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

GET /api/nsk/v2/persons/{personKey}/account/collection/{accountCollectionKey}/transactions

Summary: Retrieves the person account collection transactions based on the person key, the account collection key, and data in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
accountCollectionKey	path	string	Yes	The account collection key.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

plaintext



|

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json   
{  
    "data": {  
        "totalCount": 11085,  
        "lastPageKey": "MTAw",  
        "transactions": [  
            {  
                "accountTransactionCode": null,  
                "recordLocator": null,  
                "accountReference": null,  
                "transactionKey": "MSExITEwMjA4OCJsJNQxATUzMyEhUURRNUZ",  
                "amount": -2.42,  
                "createdDate": "2025-08-07T04:47:18.3077247Z",  
                "currencyCode": "USD",  
                "foreignAmount": -15.0,  
                "note": "Funds Used: PNR123",  
                "foreignCurrencyCode": "CNY",  
                "type": 0,  
                "paymentId": 365423,  
                "accountCollectionKey": "MSex"  
            }  
        ]  
    }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json   
{  
    "errors": [  
        {  
            "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",  
            "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
            "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
            "type": "Validation",  
            "details": {  
                "validation": "RequiredAttribute",  
                "model": "TransactionSearchRequestv2",  
                "member": "StartDate"  
            },  
            "rawMessage": "The TransactionSearchRequestv2 field is required.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
        }  
    ],  
    "data": null  
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```

json

{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when end date is before start date.

PUT /api/nsk/v1/persons/{personKey}/account/status

Summary: Updates the person account status.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
status	query	string	No	The new account status.
Enumeration values: 0 = Open, 1 = Closed, 2 = AgencyInactive, 3 = Unknown				

Responses

200 Response

500 Response

GET /api/nsk/v2/persons/{personKey}/account/transactions

Summary: Retrieves all persons account transactions.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

```

plaintext
| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional
filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string |
No | The last page key where to start paged query. |

```

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional
filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string |
No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json 
```

```
{  
  "data": {  
    "totalCount": 11085,  
    "lastPageKey": "MTAw",  
    "transactions": [  
      {  
        "accountTransactionCode": null,  
        "recordLocator": null,  
        "accountReference": null,  
        "transactionKey": "MSExITEwMjA4OCJsJNQxATUzMyEhUURRNuz",  
        "amount": -2.42,  
        "createdDate": "2025-08-07T04:47:18.306158Z",  
        "currencyCode": "USD",  
        "foreignAmount": -15.0,  
        "note": "Funds Used: PNR123",  
        "foreignCurrencyCode": "CNY",  
        "type": 0,  
        "paymentId": 365423,  
        "accountCollectionKey": "MSex"  
      }  
    ]  
  }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",  
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "type": "Validation",  
      "details": {  
        "validation": "RequiredAttribute",  
        "model": "TransactionSearchRequestv2",  
        "member": "StartDate"  
      },  
      "rawMessage": "The TransactionSearchRequestv2 field is required.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```

json

{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when end date is before start date.

GET /api/nsk/v1/persons/{personKey}/addresses

Summary: Gets the collection of addresses for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

POST /api/nsk/v1/persons/{personKey}/addresses

Summary: Creates a new address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey}

Summary: Gets a specific address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAddressKey	path	string	Yes	The unique address key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey}

Summary: Updates a specific address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The modified address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey}

Summary: Patches a specific address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The patched address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/persons/{personKey}/addresses/{personAddressKey}

Summary: Deletes a specific address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAddressKey	path	string	Yes	The unique person address key.

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/affiliations

Summary: Gets all affiliations for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

POST /api/nsk/v1/persons/{personKey}/affiliations

Summary: Creates a new affiliation for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey}

Summary: Gets a specific affiliation for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey}

Summary: Updates a specific affiliation for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Request Body

The modified affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/persons/{personKey}/affiliations/{personAffiliationKey}

Summary: Deletes a specific affiliation for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/aliases

Summary: Gets the collection of aliases for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of aliases.

POST /api/nsk/v1/persons/{personKey}/aliases

Summary: Creates a new alias for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey}

Summary: Gets a specific alias for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAliasKey	path	string	Yes	The unique person alias key.

Responses

404 Response

200 Response

The alias.

PUT /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey}

Summary: Updates a specific alias for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The modified alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey}

Summary: Patches a specific alias for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The patched alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/aliases/{personAliasKey}

Summary: Deletes a specific alias for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personAliasKey	path	string	Yes	The unique person alias key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/comments

Summary: Gets the collection of comments for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of comments.

POST /api/nsk/v1/persons/{personKey}/comments

Summary: Creates a new comment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/comments/{personCommentKey}

Summary: Gets a specific comment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personCommentKey	path	string	Yes	The unique comment key.

Responses

404 Response

200 Response

The comment.

PUT /api/nsk/v1/persons/{personKey}/comments/{personCommentKey}

Summary: Updates a specific comment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The modified comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/comments/{personCommentKey}

Summary: Patches a specific comment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The patched comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/comments/{personCommentKey}

Summary: Deletes a specific comment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personCommentKey	path	string	Yes	The unique person comment key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/emails

Summary: Gets the collection of email addresses for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of email addresses.

POST /api/nsk/v1/persons/{personKey}/emails

Summary: Creates a new email address for a specific person.

Description

The "type" property for the request is a single character. See the "/resources/EmailTypes" endpoint for valid email type codes.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey}

Summary: Gets a specific email address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personEmailAddressKey	path	string	Yes	The unique email address key.

Responses

404 Response

200 Response

The email address.

PUT /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey}

Summary: Updates a specific email address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The modified email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey}

Summary: Patches a specific email address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The patched email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/emails/{personEmailAddressKey}

Summary: Deletes a specific email address for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/information

Summary: Gets all information from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of information.

POST /api/nsk/v1/persons/{personKey}/information

Summary: Creates a new information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/information/{personInformationKey}

Summary: Gets a specific information from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personInformationKey	path	string	Yes	The unique person information key.

Responses

404 Response

200 Response

The information.

PUT /api/nsk/v1/persons/{personKey}/information/{personInformationKey}

Summary: Updates a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The modified information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/information/{personInformationKey}

Summary: Patches a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The patched information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/information/{personInformationKey}

Summary: Deletes a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personInformationKey	path	string	Yes	The unique person information key.

Responses

200 Response

Ok.

PUT /api/nsk/v1/persons/{personKey}/merge

Summary: Updates a person by merging the PNRs associated, account information and optionally the customer programs.

Description

- If new name is not given the name of the person to be updated will be retained.
- This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person to be updated and merged on to.

Request Body

The person merge request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/phoneNumbers

Summary: Gets the collection of phone numbers for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

POST /api/nsk/v1/persons/{personKey}/phoneNumbers

Summary: Creates a new phone number for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey}

Summary: Gets a specific phone number for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPhoneNumberKey	path	string	Yes	The unique phone number key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey}

Summary: Updates a specific phone number for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The modified phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey}

Summary: Patches a specific phone number for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The patched phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/persons/{personKey}/phoneNumbers/{personPhoneNumberKey}

Summary: Deletes a specific phone number for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/preferences

Summary: Gets all preferences for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of preferences.

POST /api/nsk/v1/persons/{personKey}/preferences

Summary: Creates a new preference for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey}

Summary: Gets a specific preference for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

404 Response

200 Response

The preference.

PUT /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey}

Summary: Updates a specific preference for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The modified preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey}

Summary: Patches a specific preference for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The patched preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/preferences/{personPreferenceKey}

Summary: Deletes a specific preference for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/programs

Summary: Gets the collection of programs for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of programs.

POST /api/nsk/v1/persons/{personKey}/programs

Summary: Creates a new program for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/programs/{personProgramKey}

Summary: Gets a specific program for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personProgramKey	path	string	Yes	The unique program key.

Responses

404 Response

200 Response

The program.

PUT /api/nsk/v1/persons/{personKey}/programs/{personProgramKey}

Summary: Updates a specific program for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The modified program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/persons/{personKey}/programs/{personProgramKey}

Summary: Patches a specific program for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The patched program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/programs/{personProgramKey}

Summary: Deletes a specific program for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personProgramKey	path	string	Yes	The unique person program key.

Responses

200 Response

Ok.

GET /api/nsk/v1/persons/{personKey}/storedPayments

Summary: Gets the collection of stored payments for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of stored payments.

POST /api/nsk/v1/persons/{personKey}/storedPayments

Summary: Creates a new stored payment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey}

Summary: Gets a specific stored payment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personStoredPaymentKey	path	string	Yes	The unique stored payment key.

Responses

404 Response

200 Response

The stored payment.

PATCH /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey}

Summary: Patches a specific stored payment for a specific person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The patched stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/persons/{personKey}/storedPayments/{personStoredPaymentKey}

Summary: Deletes a specific stored payment for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Responses

200 Response

Ok.

PUT /api/nsk/v2/persons/{personKey}/storedPayments/{personStoredPaymentKey}

Summary: Updates a specific stored payment for a specific person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The modified stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

GET /api/nsk/v1/persons/{personKey}/travelDocuments

Summary: Gets the collection of travel documents for a specific person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of travel documents.

POST /api/nsk/v1/persons/{personKey}/travelDocuments

Summary: Creates a new travel document for a specific person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.

Request Body

The new travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/persons/{personKey}/travelDocuments/{personTravelDocumentKey}

Summary: Gets a specific travel document for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personTravelDocumentKey	path	string	Yes	The unique travel document key.

Responses

404 Response

200 Response

The travel document.

DELETE /api/nsk/v1/persons/{personKey}/travelDocuments/{personTravelDocumentKey}

Summary: Deletes a specific travel document for a specific person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Responses

200 Response

Ok.

PUT /api/nsk/v2/persons/{personKey}/travelDocuments/{personTravelDocumentKey}

Summary: Updates a specific travel document for a specific person.

Description

Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Once the declared gender is set in the travel document, it can only be set to other values. Setting the declared gender in the request to blank or null will not cause an error in the endpoint, but the existing value will be retained.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The modified travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v2/persons/{personKey}/travelDocuments/{personTravelDocumentKey}

Summary: Patches a specific travel document for a specific person.

Description

Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Once the declared gender is set in the travel document, it can only be set to other values. Setting the declared gender in the request to blank or null will not cause an error in the endpoint, but the existing value will be retained.

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The unique person key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The patched travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/persons/{personKey}/travelNotifications

Summary: Gets person travel notifications for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.

Responses

404 Response

200 Response

POST /api/nsk/v1/persons/{personKey}/travelNotifications

Summary: Creates a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.

Request Body

The travel notification create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}

Summary: Gets a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}

Summary: Updates a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}

Summary: Patches a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The delta mapper travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}

Summary: Deletes a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events

Summary: Gets notification events for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events

Summary: Creates a notification event for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification event create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Gets a notification event for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

404 Response

DELETE

/api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Deletes a notification event from a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The notification event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

GET /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Gets notification timed events for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST /api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Creates a notification timed event for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification timed even create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

[`/api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventKey}`](#)

Summary: Gets a notification timed event for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

404 Response

PUT

[`/api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventKey}`](#)

Summary: Updates a notification timed event for a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Request Body

The updated notification timed event request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE

[/api/nsk/v1/persons/{personKey}/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventKey}](#)

Summary: Deletes a notification timed event from a person travel notification for a person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

Pricing Endpoints

POST /api/apo/v1/pricing

Summary: Creates a pricing offer in the ancillary pricing options service.

Description

GraphQL endpoints: apoPricingAdd, apoPricingAddMany

Request Details

Request Body

The price request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

GET /api/apo/v1/pricing/configuration

Summary: Gets the pricing configuration data from the ancillary pricing options service.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
source	query	string	No	The pricing configuration source.

Responses

200 Response

Promotions Endpoints

GET /api/nsk/v1/promotions

Summary: Gets promotions based on data in the request.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
PromotionCode	query	string	No	The promotion code.

plaintext



|

| OrganizationCode | query | string | No | The organization code. || EffectiveDate | query | string (date-time) | No | The effective date of the promotion. || CultureCode | query | string | No | The requested culture code. || PromotionCodeMatching | query | string | No | The matching criteria for the promotion code, by default if not set the criteria will be an exact match.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || OrganizationCodeMatching | query | string | No | The matching criteria for the organization code, by default if not set the criteria will be an exact match.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch |

Responses

200 Response

The collection of promotions, or null.

404 Response

GET /api/nsk/v1/promotions/{promotionCode}

Summary: Gets a promotion based on the promotion code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
promotionCode	path	string	Yes	The promotion code.

Responses

200 Response

The promotion, or null.

404 Response

GET /api/nsk/v1/promotions/{promotionCode}/validate

Summary: Evaluates a promotion code and optional organization code to determine if the associated promotion is valid or not.

Description

This validates the promotion code based on the logged-in user if the organization code is not provided.

Request Details

Parameters

Name	Location	Type	Required	Description
promotionCode	path	string	Yes	The promotion code.
organizationCode	query	string	No	The optional organization code.

Responses

200 Response

True if the promotion code is valid.

Queues Endpoints

GET /api/nsk/v2/queues/bookings

Summary: Gets the list of queues available.

Description

This action requires a session token. This endpoint has known issues where the total count returned is incorrect and on the first initial request, the count of the items returned is one less than the requested page size. This is an issue with the underlying NewSkies remoting call and a bug has been logged to fix these issues.

Request Details

Parameters

Name	Location	Type	Required	Description
QueueName	query	string	No	Queue name.

plaintext



|

| QueueCode | query | string | No | Queue code. || QueueCategoryCode | query | string | No | Queue category code. Please note that this should be a char and not a string. || PageSize | query | integer | No | The default page size. || LastPageIndex | query | integer (int32) | No | Used to specify the index of the last set of records retrieved when paging is performed. This is optional. |

Responses

200 Response

404 Response

DELETE /api/nsk/v1/queues/bookings/{bookingQueueCode}/items

Summary: Empties the booking queue.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueCode	path	string	Yes	The booking queue code.
subQueueCode	query	string	No	The sub-queue code.

Responses

200 Response

PUT /api/nsk/v1/queues/bookings/{bookingQueueCode}/items/{bookingQueueItemKey}

Summary: Moves to a different queue.

Description

The booking queue proxy and service model do not currently support getting the queue code the item was moved to. Therefore, it is not possible at this time to provide a link to access the item on a different queue or a configurable response containing the queue object. Additionally, the proxy and service model do not allow for getting a particular item within a queue.

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueCode	path	string	Yes	No description
bookingQueueItemKey	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

400 Response

Examples

Response 400

Booking already in queue:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:BookingAlreadyInQueue",
      "message": "nsk-server:BookingAlreadyInQueue",
      "type": "Information",
      "details": null,
      "rawMessage": "Booking is already on queue: 'DAPI', BookingID = 127160, BookingQueueItemID = 0, SegmentKey = ''.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  }
}

```

Description: Example error when trying to move a booking to another queue, and the booking is already in the target queue.

DELETE /api/nsk/v1/queues/bookings/{bookingQueueCode}/items/{bookingQueueItemKey}

Summary: Removes booking queue items from the specified queue.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueCode	path	string	Yes	No description
bookingQueueItemKey	path	string	Yes	No description

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Http response message.

GET /api/nsk/v2/queues/bookings/{bookingQueueCode}/next

Summary: Gets the next item in the booking queue.

Description

This will not delete the item from the queue. This action requires a session token.

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueCode	path	string	Yes	booking queue code to get next item from.
SubQueueCode	query	string	No	The optional sub queue code.

```

plaintext
|

```

| StartDate | query | string (date-time) | No | Begin priority date. || EndDate | query | string (date-time) | No | End priority date. || Password | query | string | No | The Password needed to access the queue. Required if queue is set to password-protected in Sky Manager. |

Responses

200 Response

404 Response

POST /api/nsk/v2/queues/bookings/items/{bookingQueueItemKey}

Summary: Places the booking queue item back on the queue.

Description

This is solely to return existing booking queue items that have been dequeued (not deleted) via GET
`/api/nsk/v2/queues/bookings/{bookingQueueCode}/next` .

Request Details

Parameters

Name	Location	Type	Required	Description
bookingQueueItemKey	path	string	Yes	The booking queue item key.

Request Body

The request object, with password needed to access the queue.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v1/queues/bookings/queueEvents/{queueEventType}

Summary: Gets the list of booking queues that match the specified event type.

Description

This action requires a session token. "Default" or "0" is not a valid queue event type and would return a 400 Bad Request.

Request Details

Parameters

Name	Location	Type	Required	Description
queueEventType	path	string	Yes	The queue event type.

Enumeration values: 0 = Default, 1 = BookingBalanceDue, 2 = BookingNegativeBalance, 3 = BookingCustomerComment, 4 = DeclinedPaymentInitial, 5 = DeclinedPaymentChange, 6 = FareOverride, 7 = ScheduleTimeChange, 8 = ScheduleTimeChangeMisconnect, 9 = ScheduleCancellation, 10 = FlightDesignatorChange, 11 = ReaccommodationMove, 12 = GdsCancelWithPendingPayment, 13 = InvalidPriceStatusOverride, 14 = FareRestrictionOverride, 15 = HeldBookings, 16 = InvalidPriceStatus, 17 = Watchlist, 18 = NonFlightServiceFee, 19 = NotAllTicketNumbersReceived, 20 = BookingSegmentOversold, 21 = ReaccommodationCancel, 22 = ExternalSsrAutoConfirmed, 23 = OpCarrierSegUpdate, 24 = OpCarrierSsrUpdate, 25 = OpCarrierOtherUpdate, 26 = NameChangeNotAllowed, 27 = InboundAscNotProcessed, 28 = OpCarrierInformationChange, 29 = BookingComponentUpdate, 30 = GroupBookings, 31 = BankDirectPnrOutOfBalance, 32 = NoSeatAssigned, 33 = SeatNumberChange, 34 = SsrNotSupportedOnNewSeat, 35 = FewerSeatPreferencesMetOnNewSeat, 36 = AosUnableToConfirmCancel, 37 = ETicketIssue, 38 = ETicketFollowup, 39 = InvoluntaryFlyAhead, 40 = ManualClearanceOnOutage, 41 = UnbalancedPoints, 42 = OpCarrierTimeChange, 43 = OaCarrierTimeChange, 44 = VoluntaryFlightChange, 45 = InvoluntaryFlightChange, 46 = HoldCancellationFailed, 47 = ScheduleTimeChangeWithDynamicQueueCode, 48 = ReaccommodationMoveWithDynamicQueueCode, 49 = ItineraryIntegrity, 50 = ReducePartyNotProcessed, 51 = CheckedPassengerUpdate, 52 = MustBeSeatGroupViolation, 57 = NameChangeWithinRule, 58 = IncompletePassengerEMDCoupon, 59 = ASVCUpdateFailed, 60 = FailedFareClassRealignment, 61 = ScheduleTimeChangeMisconnectBelowMinimum, 62 = ScheduleTimeChangeMisconnectAboveMaximum, 63 = OpCarrierMisconnect, 64 = InventoryQueuing, 65 = SelfServiceAsm, 66 = SelfServiceDelay, 67 = SelfServiceFlightCancellation, 68 = SelfServiceDrop, 69 = SelfServiceScheduleChange, 70 = SelfServiceTimeChange, 71 = AosAddOrCancelNotSuccessful, 72 = MissingGdsFareFamily, 73 = ChangeCabinQueue, 74 = PendingPaymentQueue, 75 = TicketDisplayToDetermineIncludedServiceBundleFailed, 76 = DuplicateTicketsInBooking, 77 = TicketCouponUpdateQueue, 78 = ETicketRefund

Responses

200 Response

400 Response

Examples

Response 200

With matching queues:

```
json
{
  "data": [
    {
      "queueCode": "ORATCH",
      "subQueueCode": null,
      "queueCategoryCode": "S",
      "name": "ORA Time Change",
      "totalBookings": 20,
      "restriction": 2,
      "hasPassword": false,
      "status": 0
    }
  ]
}
```

Description: Example when queues match the specified queue event type.

No matching queues:

```
json

{
    "data": []
}
```

Description: Example when no queues match the specified event type on the query parameter.

Response 400

Invalid queue event type:

```
json

{
    "errors": [
        {
            "id": "7387c8d5-d11a-f4f0-cc8b-7487d7c0990a",
            "code": "nsk:Queues:QueueEventTypeInvalid",
            "message": "nsk:Queues:QueueEventTypeInvalid",
            "type": "Validation",
            "details": null,
            "rawMessage": "Queue event type is invalid.",
            "exception": null,
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
        }
    ],
    "messages": [
        {
            "rawValue": "The value 'kk' is not valid.",
            "code": "core:Validation:MalformedRequest",
            "debugMode": "The API is currently running in debug mode. In production the 'rawValue' node will not be returned. These are returned for informational purposes only.",
            "type": "Information",
            "value": "core:Validation:MalformedRequest",
            "status": 1,
            "details": null
        }
    ],
    "data": null
}
```

Description: Example when the queue event type is invalid.

Default queue event type:

```
json

{
    "errors": [
        {
            "id": "2a229cf0-263b-78cd-4233-43ad8a1d1171",
            "code": "nsk:Queues:QueueEventTypeInvalid",
            "message": "nsk:Queues:QueueEventTypeInvalid",
            "type": "Validation",
            "details": null,
            "rawMessage": "Queue event type is invalid.",
            "exception": null,
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
        }
    ],
    "data": null
}
```

Description: Example when the queue event type is Default or 0.

POST /api/nsk/v1/queues/travel

Summary: Creates a new travel queue entry.

Description

This action requires a session token.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/queues/travel/{travelQueueCode}/next

Summary: Pops the next item in the travel queue.

Description

This will delete the item from the queue. This action requires a session token.

Request Details

Parameters

Name	Location	Type	Required	Description
travelQueueCode	path	string	Yes	Travel queue code to get next item from.
subQueueCode	query	string	No	Sub queue code.

Responses

200 Response

Returned successfully.

404 Response

Redis Endpoints

GET /api/v1/redis

Summary: Gets all the cache items that exist in redis.

Description

Gets all the cache items that exist in redis.

Request Details

Responses

200 Response

OK. Request completed successfully.

401 Response

Your token is unauthorized to use this endpoint.

400 Response

A bad request was submitted.

Examples

Response 200

Ok:

```
json   
{  
  "data": [  
    "Mac",  
    "FlightSearchSettings",  
    "Station"  
  ]  
}
```

Description: Data results are based on the current cache.

Response 401

Not Allowed:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "core:Caching:CacheManagementNotAllowed",  
      "message": null,  
      "type": "Error",  
      "details": null,  
      "rawMessage": "The current token's permissions are not allowed to manage the redis cache.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Role permissions is not allowed. Please see the developer documentation portal for more information.

Response 400

Not Enabled:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "core:Caching:CacheManagementNotAllowed",  
      "message": null,  
      "type": "Error",  
      "details": null,  
      "rawMessage": "The ability to manage redis cache is not enabled. Please change the 'ManagementEnabled' flag in the 'RedisConnection' configuration section of your application's configuration file.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Please, log a SNOW request to enable this ability.

DELETE /api/v1/redis

Summary: Deletes all cache items.

Description

This will ignore all session values. If an in memory cache is configured via "cacheOptions.serverCache" (default is 5 minutes) other API instances may still have locally stale cached data for the configured time.

Request Details

Responses

200 Response

OK. Request completed successfully.

401 Response

Your token is unauthorized to use this endpoint.

400 Response

A bad request was submitted.

Examples

Response 200

Ok:

```
json
{
  "data": null
}
```

Description: No content returned.

Response 401

Not Allowed:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "core:Caching:CacheManagementNotAllowed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "The current token's permissions are not allowed to manage the redis cache.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Role permissions is not allowed. Please see the developer documentation portal for more information.

Response 400

Resource Not Found:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "core:Validation:ResourceNotFound",
      "message": null,
      "type": "Error",
      "details": {
        "name": "{parameter name}",
        "value": "{parameter value}"
      },
      "rawMessage": "The data you are trying to access is not valid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}

```

Description: The value was not found.

DELETE /api/v1/redis/{name}

Summary: Deletes a specific cache item from redis based on the cache item name.

Description

If an in memory cache is configured via "cacheOptions.serverCache" (default is 5 minutes) other API instances may still have locally stale cached data for the configured time.

Request Details

Parameters

Name	Location	Type	Required	Description
name	path	string	Yes	The name of the cache item.

Responses

200 Response

OK. Request completed successfully.

401 Response

Your token is unauthorized to use this endpoint.

400 Response

A bad request was submitted.

Examples

Response 200

Ok:

```

json

{
  "data": null
}

```

Description: No content returned.

Response 401

Not Allowed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "core:Caching:CacheManagementNotAllowed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "The current token's permissions are not allowed to manage the redis cache.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Role permissions is not allowed. Please see the developer documentation portal for more information.

Response 400

Required Argument:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "core:Validation:RequiredArgument",
      "message": null,
      "type": "Error",
      "details": {
        "argument": "{argument}"
      },
      "rawMessage": "The request is missing a required argument expected in the request.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Check url arguments.

Resource Not Found:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "core:Validation:ResourceNotFound",
      "message": null,
      "type": "Error",
      "details": {
        "name": "{parameter name}",
        "value": "{parameter value}"
      },
      "rawMessage": "The data you are trying to access is not valid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Check url argument.

Resources Endpoints

GET /api/nsk/v1/resources/AccountTransactionCodes

Summary: Retrieves the collection of account transaction code resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/AccountTransactionCodes/{transactionCode}

Summary: Retrieves the specific account transaction code resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
transactionCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/AddressTypes

Summary: Retrieves the collection of address type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/AddressTypes/{addressTypeCode}

Summary: Retrieves the specific address type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
addressTypeCode	path	string	Yes	The unique type code. Please note that this should be a char and not a string.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Areas

Summary: Retrieves the collection of area resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Areas/{areaCode}

Summary: Retrieves the specific area resource.

Description

See `/api/nsk/v1/resources/areas/{areaCode}/zones` to retrieve the list of geographical zones associated with a specific area.

Request Details

Parameters

Name	Location	Type	Required	Description
areaCode	path	string	Yes	The unique area code.
cultureCode	query	string	No	The unique culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Areas/{areaCode}/zones

Summary: Retrieves the collection of zones resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
areaCode	path	string	Yes	The unique code of the area to which the zones belong.
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Areas/{areaCode}/zones/{zoneCode}

Summary: Retrieves the specific zone resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
areaCode	path	string	Yes	The unique code of the area to which the zone belongs.
zoneCode	path	string	Yes	The unique zone code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/BaggageTypes

Summary: Retrieves the collection of baggage type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve baggage types response:

```
json   
{  
    "data": [  
        {  
            "baggageTypeCode": "1BAG",  
            "bagWeightType": 2,  
            "freeBag": false,  
            "inActive": false,  
            "name": "Booking Baggage Allowance",  
            "weightReducedBy": 5,  
            "includeInBaggageCount": false  
        },  
        {  
            "baggageTypeCode": "BSMX",  
            "bagWeightType": 2,  
            "freeBag": false,  
            "inActive": false,  
            "name": "BSM ExceptionTesting",  
            "weightReducedBy": 0,  
            "includeInBaggageCount": false  
        },  
        {  
            "baggageTypeCode": "BSM3",  
            "bagWeightType": 1,  
            "freeBag": false,  
            "inActive": false,  
            "name": "BSM Exception Testing",  
            "weightReducedBy": 0,  
            "includeInBaggageCount": false  
        },  
        {  
            "baggageTypeCode": "MUSIC",  
            "bagWeightType": 1,  
            "freeBag": false,  
            "inActive": false,  
            "name": "Musical Equipment",  
            "weightReducedBy": 5,  
            "includeInBaggageCount": false  
        },  
        {  
            "baggageTypeCode": "BSME",  
            "bagWeightType": 0,  
            "freeBag": true,  
            "inActive": false,  
            "name": "OPS Baggage - DO NOT EDIT",  
            "weightReducedBy": 0,  
            "includeInBaggageCount": false  
        },  
        {  
            "baggageTypeCode": "MEDIC",  
            "bagWeightType": 0,  
            "freeBag": true,  
            "inActive": false,  
            "name": "Medical Equipment",  
            "weightReducedBy": 0,  
            "includeInBaggageCount": false  
        }  
    ]  
}
```

Description: Example response for successfully retrieving the configured baggage types in the system.

GET /api/nsk/v1/resources/BaggageTypes/{baggageTypeCode}

Summary: Retrieves the specific baggage type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
baggageTypeCode	path	string	Yes	The unique baggage type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/bundles

Summary: Retrieves the collection of bundle configuration resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/bundles/{bundleCode}

Summary: Retrieves a specific bundle configuration based on the bundle code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
bundleCode	path	string	Yes	The bundle code.
cultureCode	query	string	No	The culture code.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/bundles/applications

Summary: Retrieves the collection of bundle application resources.

Description

See `api/nsk/v1/bundles/application/{bundleApplicationKey}/details` to retrieve additional bundle application details. The application details consist of three lists: carrier codes, product classes, and classes of service.

Request Details

Parameters

Name	Location	Type	Required	Description
BundleRuleCode	query	string	No	The bundle rule code.

```
plaintext 
```

| CultureCode | query | string | No | The specific culture code. || ETag | query | string | No | The cache eTag for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/bundles/rules

Summary: Retrieves the collection of bundle rule resources.

Description

See `api/nsk/v1/bundles/rule/{bundleRuleCode}/details` to retrieve a list of bundle applications associated with a specific bundle rule. Bundle rules are used to control where and how bundle applications are used.

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/bundles/sets

Summary: Retrieves the collection of bundle set resources.

Description

See `api/nsk/v1/bundles/set/{bundleSetCode}/details` to retrieve a list of bundle codes associated with a specific bundle set.

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/bundles/ssrs

Summary: Retrieves the collection of bundle ssr configurations associated with the bundle code.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
BundleCode	query	string	No	The bundle code.

plaintext



|

| CultureCode | query | string | No | The specific culture code. || ETag | query | string | No | The cache eTag for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/carriers

Summary: Retrieves the collection of carrier resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

[404 Response](#)

[304 Response](#)

[200 Response](#)

GET /api/nsk/v1/resources/carriers/{carrierCode}

Summary: Retrieves a single carrier.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
carrierCode	path	string	Yes	The unique carrier code.
cultureCode	query	string	No	The specific culture code.
eTag	query	string	No	The eTag.

Responses

[404 Response](#)

[200 Response](#)

GET /api/nsk/v1/resources/Cities

Summary: Retrieves the collection of city resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

[404 Response](#)

[304 Response](#)

[200 Response](#)

GET /api/nsk/v1/resources/Cities/{cityCode}

Summary: Retrieves the specific city resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
cityCode	path	string	Yes	The unique city code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ClassOfServices

Summary: Retrieves the collection of class of service resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext Copy
|
| CultureCode | query | string | No | The desired culture code to be returned. |
| ETag | query | string | No | The cache token for this request. Data will
not be returned if etag still matches. |
| StartIndex | query | integer (int32) | No | The start index value for this request. |
| ItemCount | query | integer (int32) | No | The item count value for this request. |
```

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ClassOfServices/{classOfServiceCode}

Summary: Retrieves the resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
classOfServiceCode	path	string	Yes	The unique class of service code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/CommissionRates

Summary: Retrieves the collection of commission rate resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext | 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/CommissionRates/{commissionRateCode}

Summary: Retrieves the specific commission rate resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
commissionRateCode	path	string	Yes	The unique commission rate code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ContactTypes

Summary: Retrieves the collection of contact type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext | 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

[404 Response](#)

[304 Response](#)

[200 Response](#)

[GET /api/nsk/v1/resources/ContactTypes/{contactTypeCode}](#)

Summary: Retrieves the specific contact type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
contactTypeCode	path	string	Yes	The unique contact type code. Please note that this should be a char and not a string.
cultureCode	query	string	No	The specific culture code.

Responses

[404 Response](#)

[200 Response](#)

[GET /api/nsk/v1/resources/contents](#)

Summary: Searches the contents based on the search criteria provided.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
Name	query	string	No	Name of the content.

plaintext 

| Type | query | string | No | Specific content type.

Enumeration values: 0 = GeneralReference, 1 = News, 2 = ConsoleHelp, 3 = FareRuleReference, 4 = ReviewWithCustomer, 8 = Image, 9 = PromotionReference, 10 = SubscriptionReference || ContainerId | query | integer (int64) | No | Unique id of the container. || PageSize | query | integer | No | The default page size. || LastContentId | query | integer (int64) | No | Used to specify the last item of the page of the last set of records retrieved when paging was performed. This is optional. |

Responses

[200 Response](#)

Search content succeeded.

[GET /api/nsk/v1/resources/contents/{contentId}](#)

Summary: Retrieves a resource setting and its related information.

Description

This is a post to easily allow special path characters. See /resources/settings/data to avoid manually parsing the data.

Request Details

Parameters

Name	Location	Type	Required	Description
contentId	path	integer (int64)	Yes	The content id.
convertRtfToHtml	query	boolean	No	The bool to determine whether or not to convert RTF to HTML data.
eTag	query	string	No	ETag for caching.

Responses

404 Response

304 Response

200 Response

DELETE /api/nsk/v1/resources/contents/{contentId}

Summary: Deletes a content item.

Description

This endpoint currently only supports deleting news content items. Deletion of other content types will be supported in the future.

Request Details

Parameters

Name	Location	Type	Required	Description
contentId	path	integer (int64)	Yes	The content identifier.

Responses

400 Response

There was a problem with the request and it could not be completed.

200 Response

OK. Request completed successfully.

Examples

Response 400

Deleting content is not allowed:

```
json   
{  
  "errors": [  
    {  
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",  
      "code": "nsk:Content>DeleteContentNotAllowed",  
      "message": "nsk:Content>DeleteContentNotAllowed",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "Deleting content is not allowed.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example when deleting content is not allowed.

Deleting news is not allowed:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:Content>DeleteNewsNotAllowed",
      "message": "nsk:Content>DeleteNewsNotAllowed",
      "type": "Validation",
      "details": null,
      "rawMessage": "Deleting news is not allowed.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when deleting news is not allowed.

Content does not exist:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:Content>ContentDoesNotExist",
      "message": "nsk:Content>ContentDoesNotExist",
      "type": "Validation",
      "details": null,
      "rawMessage": "The content does not exist.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the content does not exist.

Deleting non-news content:

```
json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "nsk:Content>DeleteNonNewsContentNotSupported",
      "message": "nsk:Content>DeleteNonNewsContentNotSupported",
      "type": "Validation",
      "details": null,
      "rawMessage": "Deleting non-news content is currently not supported.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example when the content is not a news. Non-news content deletion is currently not supported.

Content ID is null.:

```

json

{
  "errors": [
    {
      "id": "ebaf3f43-1fd2-7c9b-5f3c-1d0778e52701",
      "code": "core:Validation:NullRequest",
      "message": "The request is null or couldn't be converted to the expected data type.",
      "type": "Error",
      "details": null,
      "rawMessage": "The request is null or couldn't be converted to the expected data type.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    "data": null
  }
}

```

Description: Example when the content ID is null. It can be set to null by setting it to a URL encoded space value %20 or to a value that cannot be converted to type long .

Response 200

Content deleted successfully:

```

json

{
  "data": null
}

```

Description: Example when the content deleted successfully.

GET /api/nsk/v1/resources/contents/{contentId}/data

Summary: Retrieves resource setting data based on the accept content type.

Description

This is a post to easily allow special path characters. If the data returned is not in HTML format and convertRtfToHtml is set to true, the content requested was of an incompatible data type.

Request Details

Parameters

Name	Location	Type	Required	Description
contentId	path	integer (int64)	Yes	The content id.
convertRtfToHtml	query	boolean	No	The bool to determine whether or not to convert RTF to HTML data.
eTag	query	string	No	ETag for caching.

Responses

404 Response

304 Response

200 Response

An HTTP response with a list of Domains that match the given parameters.

GET /api/nsk/v2/resources/countries

Summary: Retrieves the collection of country resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/countries/{countryCode}

Summary: Retrieves a specific country resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
countryCode	path	string	Yes	The unique country code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/cultures

Summary: Retrieves the resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/cultures/{cultureCode}

Summary: Retrieves the resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
cultureCode	path	string	Yes	The unique culture code.
forCultureCode	query	string	No	The for culture.

Responses

404 Response

200 Response

GET /api/nsk/v2/resources/currencies

Summary: Retrieves the collection of currency resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/currencies/{currencyCode}

Summary: Retrieves the specific currency resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
currencyCode	path	string	Yes	The unique currency code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v2/resources/customerPrograms

Summary: Retrieves the collection of customer program resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext Copy
|
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve Customer Programs v2 Response:

```

json
{
  "data": {
    "programCode": "ABC",
    "name": "ABC Customer Program",
    "programType": 1,
    "inActive": false,
    "customerProgramLevels": [
      {
        "programCode": "ABC",
        "programLevelCode": "LV1",
        "name": "ABC Level 1",
        "shortName": "ABC Level 1",
        "displayCode": "1",
        "rank": 1,
        "inActive": false,
        "customerProgramSettings": [
          {
            "programCode": "NV",
            "programLevel": "NV1",
            "settingCode": "BoardingZone",
            "sequenceNumber": 1,
            "settingValue": "1"
          }
        ]
      }
    ],
    "checkDigitType": 1
  }
}

```

Description: Example response for successfully retrieving the customer programs v2 resources

GET /api/nsk/v2/resources/customerPrograms/{programCode}

Summary: Retrieves the specific customer program resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
programCode	path	string	Yes	The unique program code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

Examples

Response 200

Retrieve Customer Programs v2 Response:

```

json
{
  "data": {
    "programCode": "ABC",
    "name": "ABC Customer Program",
    "programType": 1,
    "inActive": false,
    "customerProgramLevels": [
      {
        "programCode": "ABC",
        "programLevelCode": "LV1",
        "name": "ABC Level 1",
        "shortName": "ABC Level 1",
        "displayCode": "1",
        "rank": 1,
        "inActive": false,
        "customerProgramSettings": [
          {
            "programCode": "NV",
            "programLevel": "NV1",
            "settingCode": "BoardingZone",
            "sequenceNumber": 1,
            "settingValue": "1"
          }
        ]
      }
    ],
    "checkDigitType": 1
  }
}

```

Description: Example response for successfully retrieving the customer programs v2 resources

GET /api/nsk/v1/resources/CustomerPrograms/{programCode}/levels/{programLevelCode}

Summary: Retrieves the specific program level.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
programCode	path	string	Yes	The unique program code.
programLevelCode	path	string	Yes	The unique program level code.
cultureCode	query	string	No	The culture code.

Responses

404 Response

304 Response

200 Response

The customer program level.

GET /api/nsk/v1/resources/Delays

Summary: Retrieves the collection of delay resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Delays/{delayCode}

Summary: Retrieves the specific delay resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
delayCode	path	string	Yes	The unique delay code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Departments

Summary: Retrieves the collection of department resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Departments/{departmentCode}

Summary: Retrieves the specific department resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
departmentCode	path	string	Yes	The unique department code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/distributionOptions

Summary: Retrieves the localizations.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CultureCode	query	string	No	The desired culture code to be returned.

plaintext 

|

| ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No
| The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/DocumentTypes

Summary: Retrieves the collection of document type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/DocumentTypes/{documentTypeCode}

Summary: Retrieves the specific document type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
documentTypeCode	path	string	Yes	The unique document type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Domains

Summary: Retrieves the collection of domain resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Domains/{domainCode}

Summary: Retrieves the specific domain resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
domainCode	path	string	Yes	The unique domain code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/EmailTypes

Summary: Retrieves the collection of email type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/EmailTypes/{emailTypeCode}

Summary: Retrieves the specific email type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
emailTypeCode	path	string	Yes	The unique email type code. Please note that this should be a char and not a string.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ExternalRates

Summary: Retrieves the collection of external rate resources.

Description

The external rates cache expiration is configurable as a number of minutes past the hour to expire. This should allow for better alignment with third party external rates providers' recommendations, which typically state that external rates cache items should expire 5-7 minutes past the hour. This can be configured in the NskSystemOptions.ExternalRatesExpiration. If no value is set, the API will default to five minutes past the hour. If the value "Enable External Currency Conversion" is disabled in SkyPay, external rates won't be returned by this call. Additionally, there are some edge cases where the cache entry expiration is created at or close to the configured expiration time. In these cases, an additional two minutes will be added to the cache expiration. For example, if the configured external rate cache expiration is for five minutes after the hour, and the current time is 12:04, the expiration will be 12:07 instead of 12:05. This gives the cached item enough time to get set into cache, but will expire quickly in order to refresh according to third party external rates providers' cache requirements (refresh at 5-7 minutes past the hour).

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ExternalRates/{ratelId}

Summary: Retrieves the specific external rate resource.

Description

The external rates cache expiration is configurable as a number of minutes past the hour to expire. This should allow for better alignment with third party external rates providers' recommendations, which typically state that external rates cache items should expire 5-7 minutes past the hour. This can be configured in the NskSystemOptions.ExternalRatesExpiration. If no value is set, the API will default to five minutes past the hour. If the value "Enable External Currency Conversion" is disabled in SkyPay, external rates won't be returned by this call.

Request Details

Parameters

Name	Location	Type	Required	Description
ratelId	path	string	Yes	The unique rate ID.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ExternalSystems

Summary: Retrieves the collection of external system resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/externalSystems/{systemCode}

Summary: Retrieves a single external system resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
systemCode	path	string	Yes	The unique external system code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/fareBasisCodes

Summary: Retrieves the list of fare basis codes.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	No description

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Fare basis response:

```
json
[ "A", "A0807", "A2FARE", "AAAAFARE" ]
```

Description: Gets the list of fare basis codes

GET /api/nsk/v1/resources/fareDiscounts

Summary: Retrieves a collection of fare discount resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext
|
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/fareDiscounts/{fareDiscountCode}

Summary: Retrieves a specific fare discount resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
fareDiscountCode	path	string	Yes	The unique fare discount code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/fareSurcharges

Summary: Retrieves a collection of fare surcharge resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/fareSurcharges/{fareSurchargeCode}

Summary: Retrieves a specific fare surcharge resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
fareSurchargeCode	path	string	Yes	The unique fare surcharge code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/FareTypes

Summary: Retrieves the collection of fare type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/FareTypes/{fareTypeCode}

Summary: Retrieves the specific fare type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
fareTypeCode	path	string	Yes	The unique fare type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Fees

Summary: Retrieves the collection of fee resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Fees/{feeCode}

Summary: Retrieves the specific fee resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
feeCode	path	string	Yes	The unique fee code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/fees/{feeCode}/details

Summary: Gets the fee options.

Description

Note, this endpoint is not cached like other resource endpoints. The NSK enumeration "TSI.GroundTimeBeforeDepartureFromConnectionWithin" is synonymous with "TravelSegmentIndicator.DepartureWithinStopoverTime". The NSK enumeration "TSI.GroundTimeBeforeDepartureFromConnectionExceeds" is synonymous with "TravelSegmentIndicator.DepartureExceedsStopoverTime".

Request Details

Parameters

Name	Location	Type	Required	Description
feeCode	path	string	Yes	The unique fee code.

Responses

200 Response

The fare rules.

404 Response

GET /api/nsk/v1/resources/genders

Summary: Retrieves the localizations.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CultureCode	query	string	No	The desired culture code to be returned.

```
plaintext ✖
|
```

| ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. | | StartIndex | query | integer (int32) | No
| The start index value for this request. | | ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/HotCardReasons

Summary: Retrieves the collection of hot card reason resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/HotCardReasons/{hotCardReasonCode}

Summary: Retrieves the specific hot card reason resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
hotCardReasonCode	path	string	Yes	The unique hot card reason.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/IssuanceReasons

Summary: Retrieves the collection of issuance reason resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/IssuanceReasons/{issuanceReasonCode}

Summary: Retrieves the specific issuance reason resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
issuanceReasonCode	path	string	Yes	The unique issuance reason code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Locations

Summary: Retrieves the collection of location resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Locations/{locationCode}

Summary: Retrieves the specific location resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
locationCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v2/resources/macs

Summary: Retrieves the collection of mac resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Macs/{macCode}

Summary: Retrieves the specific mac resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
macCode	path	string	Yes	The unique mac code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/marketRoutes

Summary: Retrieves the collection of market route resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
DepartureStation	query	string	No	Gets or sets the departure station for this request.

```
plaintext 
```

|

| ArrivalStation | query | string | No | Gets or sets the arrival station for this request. || ETag | query | string | No | The cache eTag for this request. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

304 Response

400 Response

200 Response

Examples

Response 304

Matching ETag:

```
json 
```

{
 "data": null
}

Description: Response when an ETag is provided and the cached data still matches. No Market Routes are returned regardless of the station filters specified.

Response 400

Invalid Departure Station Length Error:

```
json 
```

{
 "errors": [
 {
 "id": "2f5dd7f3-47f7-48fb-a180-f9152278605f",
 "code": "MarketRoutesQuery.DepartureStation:StringLengthAttribute",
 "message": "MarketRoutesQuery.DepartureStation:StringLengthAttribute",
 "type": "Validation",
 "details": {
 "validation": "StringLengthAttribute",
 "model": "MarketRoutesQuery",
 "member": "DepartureStation"
 },
 "rawMessage": "The field MarketRoutesQuery must be a string with a maximum length of 3.",
 "exception": null,
 "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
 }
],
 "data": null
}

Description: Example error when departure station is an incorrect length.

Invalid Arrival Station Length Error:

```
json
{
  "errors": [
    {
      "id": "2f5dd7f3-47f7-48fb-a180-f9152278605f",
      "code": "MarketRoutesQuery.ArrivalStation:StringLengthAttribute",
      "message": "MarketRoutesQuery.ArrivalStation:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "MarketRoutesQuery",
        "member": "ArrivalStation"
      },
      "rawMessage": "The field MarketRoutesQuery must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when arrival station is an incorrect length.

Response 200

Default without DepartureStation or ArrivalStation filters:

json



```
{
  "data": [
    {
      "arrivalStation": "BOS",
      "departureStation": "ATL",
      "connectionMarketSegments": [
        {
          "departureStation": "ATL",
          "arrivalStation": "BOS",
          "segmentNumber": 1
        }
      ]
    },
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "ATL",
          "segmentNumber": 2
        },
        {
          "departureStation": "ATL",
          "arrivalStation": "JFK",
          "segmentNumber": 3
        }
      ]
    },
    {
      "arrivalStation": "JFK",
      "departureStation": "DEN",
      "connectionMarketSegments": [
        {
          "departureStation": "DEN",
          "arrivalStation": "JFK",
          "segmentNumber": 1
        }
      ]
    },
    {
      "arrivalStation": "BOS",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "BOS",
          "segmentNumber": 2
        }
      ]
    },
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        }
      ]
    }
  ]
}
```

```

        "departureStation": "DEN",
        "arrivalStation": "JFK",
        "segmentNumber": 2
    }
]
}
]
}

```

Description: Response when neither DepartureStation or ArrivalStation are specified. All Market Routes are returned.

DepartureStation filter specified:

```

json
{
  "data": [
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "ATL",
          "segmentNumber": 2
        },
        {
          "departureStation": "ATL",
          "arrivalStation": "JFK",
          "segmentNumber": 3
        }
      ]
    },
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "JFK",
          "segmentNumber": 2
        }
      ]
    }
  ]
}

```

Description: Response when DepartureStation filter specified but not ArrivalStation. Market Routes with the specified DepartureStation are returned.

ArrivalStation filter specified:

```
json 
```

```
{  
    "data": [  
        {  
            "arrivalStation": "BOS",  
            "departureStation": "SLC",  
            "connectionMarketSegments": [  
                {  
                    "departureStation": "SLC",  
                    "arrivalStation": "DEN",  
                    "segmentNumber": 1  
                },  
                {  
                    "departureStation": "DEN",  
                    "arrivalStation": "BOS",  
                    "segmentNumber": 2  
                }  
            ]  
        },  
        {  
            "arrivalStation": "BOS",  
            "departureStation": "ATL",  
            "connectionMarketSegments": [  
                {  
                    "departureStation": "ATL",  
                    "arrivalStation": "BOS",  
                    "segmentNumber": 1  
                }  
            ]  
        }  
    ]  
}
```

Description: Response when ArrivalStation filter specified but not DepartureStation. Market Routes with the specified ArrivalStation are returned.

DepartureStation and ArrivalStation filters specified:

```

json

{
  "data": [
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "ATL",
          "segmentNumber": 2
        },
        {
          "departureStation": "ATL",
          "arrivalStation": "JFK",
          "segmentNumber": 3
        }
      ]
    },
    {
      "arrivalStation": "JFK",
      "departureStation": "SLC",
      "connectionMarketSegments": [
        {
          "departureStation": "SLC",
          "arrivalStation": "DEN",
          "segmentNumber": 1
        },
        {
          "departureStation": "DEN",
          "arrivalStation": "JFK",
          "segmentNumber": 2
        }
      ]
    }
  ]
}

```

Description: Response when both DepartureStation and ArrivalStation filters are specified. Only Market Routes with the specified DepartureStation and ArrivalStation are returned.

GET /api/nsk/v2/resources/markets

Summary: Retrieves the collection of market resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```

plaintext
| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will
not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer
(int32) | No | The item count value for this request. |

```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/markets/{locationCode}

Summary: Retrieves a specific market resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
locationCode	path	string	Yes	The unique location code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v2/resources/markets/{locationCode}/{travelLocationCode}

Summary: Retrieves the resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
locationCode	path	string	Yes	No description
travelLocationCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/MessageTypes

Summary: Retrieves the collection of message type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/MessageTypes/{messageTypeCode}

Summary: Retrieves the specific message type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
messageTypeCode	path	string	Yes	The unique message type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/organizationStatuses

Summary: Retrieves the localizations.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CultureCode	query	string	No	The desired culture code to be returned.

plaintext	

| ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PassengerDiscounts

Summary: Retrieves the collection of passenger discount resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PassengerDiscounts/{passengerDiscountCode}

Summary: Retrieves the specific passenger discount resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerDiscountCode	path	string	Yes	The unique passenger discount code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/PassengerTypes

Summary: Retrieves the collection of passenger type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PassengerTypes/{passengerTypeCode}

Summary: Retrieves the specific passenger type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
passengerTypeCode	path	string	Yes	The unique passenger type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/PaymentMethods

Summary: Retrieves the collection of payment method resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. || currencyCode | query | string | No | The currency code. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PaymentMethods/{paymentMethodCode}

Summary: Retrieves the specific payment method resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
paymentMethodCode	path	string	Yes	The unique payment method code.
cultureCode	query	string	No	The specific culture code.
currencyCode	query	string	No	The specific currency code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/PersonInformationTypes

Summary: Retrieves the collection of person information type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext   
|
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PersonInformationTypes/{personInformationTypeCode}

Summary: Retrieves the specific person information type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personInformationTypeCode	path	string	Yes	The unique person information type code. Please note that this should be a char and not a string.

```
plaintext   
|
```

| cultureCode | query | string | No | The specific culture code. |

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/PhoneTypes

Summary: Retrieves the collection of phone number type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PhoneTypes/{phoneTypeCode}

Summary: Retrieves the specific phone number type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
phoneTypeCode	path	string	Yes	The unique phone type. Please note that this should be a char and not a string.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/postalCodes

Summary: Retrieves the collection of postal codes resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CountryCode	query	string	No	The country code.

```
plaintext 
```

| ProvinceStateCode | query | string | No | The province state code. || City | query | string | No | The city. || PageSize | query | integer | No | The default page size. || LastPageKey | query | string | No | Used to specify the last item of the page of the last set of records retrieved when paging was performed. This is optional. || PostalCode | query | string | No | The postal code. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/postalCodes/{postalCode}

Summary: Retrieves the postal code resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
postalCode	path	string	Yes	The unique postal code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/printers

Summary: Retrieves a collection of printer resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CultureCode	query	string	No	The desired culture code to be returned.

```
plaintext 
```

| ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/printers/{printerCode}

Summary: Retrieves a specific printer resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
printerCode	path	string	Yes	The unique printer code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ProductClasses

Summary: Retrieves the collection of product class resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ProductClasses/{productClassCode}

Summary: Retrieves the specific product class resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
productClassCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/PromotionSettingTypes

Summary: Retrieves the collection of promotion setting type resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/PromotionSettingTypes/{promotionSettingsTypeCode}

Summary: Retrieves the specific promotion setting type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
promotionSettingsTypeCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ProvinceStates

Summary: Retrieves the collection of province state resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ProvinceStates/{provinceStateCode}

Summary: Retrieves the specific province state resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
provinceStateCode	path	string	Yes	The unique province state code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/queue/categories

Summary: Gets a response with a list of QueueCategories that match the given parameters.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

An HTTP response with a list of QueueCategories that match the given parameters.

GET /api/nsk/v1/resources/queue/categories/{queueCategoryCode}

Summary: Gets a response with a single QueueCategory that match the given parameters.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
queueCategoryCode	path	string	Yes	Queue Category Code being requested. Please note that this should be a char and not a string.

```
plaintext 
```

| cultureCode | query | string | No | [Optional] The culture code for the resources. |

Responses

200 Response

An HTTP response with a single QueueCategory that match the given parameters.

404 Response

GET /api/nsk/v1/resources/reminderTimes

Summary: Retrieves the collection of reminder time resource that contains valid times for all passenger travel notification timed events.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/reminderTimes/{timedEventType}

Summary: Retrieves a specific reminder time resource that contains valid times for a passenger travel notification timed event.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
timedEventType	path	string	Yes	The timed event type. Enumeration values: 0 = Departure, 1 = Arrival

Responses

404 Response

200 Response

GET /api/nsk/v2/resources/roles

Summary: Retrieves the role resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/roles/{roleCode}

Summary: Retrieves a single role resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
roleCode	path	string	Yes	The unique role code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/roles/{roleCode}/settings

Summary: Retrieves the resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
roleCode	path	string	Yes	The unique role code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/RoleSettingTypes

Summary: Retrieves the collection of role setting resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext   
|
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/RoleSettingTypes/{roleSettingTypeCode}

Summary: Retrieves the specific role setting resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
roleSettingTypeCode	path	string	Yes	The unique role setting type code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/searchTypes

Summary: Retrieves the localizations.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
CultureCode	query	string	No	The desired culture code to be returned.

```
plaintext   
|
```

| ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/seatPropertyTypes

Summary: Retrieves a list of seat property types.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/resources/seatPropertyTypes/{seat.PropertyTypeCode}

Summary: Retrieves a single seat property type resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
seat.PropertyTypeCode	path	string	Yes	The unique seat property type code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

POST /api/nsk/v1/resources/settings

Summary: Retrieves a resource setting and its related information.

Description

This is a post to easily allow special path characters. See /resources/settings/data to avoid manually parsing the data.

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	ETag for caching.

Request Body

The resource settings request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

404 Response

304 Response

200 Response

POST /api/nsk/v1/resources/settings/data

Summary: Retrieves resource setting data based on the accept content type.

Description

This is a post to easily allow special path characters.

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	ETag for caching.

Request Body

The resource settings request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

404 Response

304 Response

415 Response

200 Response

An HTTP response with a list of Domains that match the given parameters.

GET /api/nsk/v1/resources/ShoppingAttributeGroups

Summary: Retrieves the collection of shopping attribute group resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ShoppingAttributeGroups/{shoppingAttributeGroupCode}

Summary: Retrieves the specific shopping attribute group resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
shoppingAttributeGroupCode	path	string	Yes	The unique shopping attribute group code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/ShoppingAttributes

Summary: Retrieves the collection of shopping attribute resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/ShoppingAttributes/{shoppingAttributeCode}

Summary: Retrieves the specific shopping attribute resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
shoppingAttributeCode	path	string	Yes	The unique shopping attribute code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/singleSignOnProvider

Summary: Retrieves a list of available single sign on providers.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/singleSignOnProvider/{providerName}

Summary: Retrieves a specific single sign on provider.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerName	path	string	Yes	The single sign on provider name.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/SsrGroups

Summary: Retrieves the collection of ssr group resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/SsrGroups/{ssrGroupCode}

Summary: Retrieves the specific ssr group resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ssrGroupCode	path	string	Yes	No description
cultureCode	query	string	No	No description

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/SsrNests

Summary: Retrieves the collection of ssr nest resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/SsrNests/{ssrNestCode}

Summary: Retrieves the specific ssr nest resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ssrNestCode	path	string	Yes	The unique ssr nest code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/SsrRestrictionResults

Summary: Retrieves the collection of ssr restriction result resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/SsrRestrictionResults/{ssrRestrictionResultCode}

Summary: Retrieves the specific ssr restriction result. resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ssrRestrictionResultCode	path	string	Yes	The unique ssr restriction result code.
cultureCode	query	string	No	The unique culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Ssrs

Summary: Retrieves the collection of ssr resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Ssrs/{ssrCode}

Summary: Retrieves the specific ssr resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ssrCode	path	string	Yes	The unique ssr code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/StandByPriorities

Summary: Retrieves the collection of standby priority resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

```
plaintext 
```

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/StandByPriorities/{standByPriorityCode}

Summary: Retrieves the specific standby priority resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
standByPriorityCode	path	string	Yes	The unique standby priority code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/StationCategories

Summary: Retrieves the collection of station category resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/StationCategories/{stationCategoryCode}

Summary: Retrieves the specific station category resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
stationCategoryCode	path	string	Yes	The unique station category code. Please note that this should be a char and not a
				string.

plaintext



| cultureCode | query | string | No | The specific culture code. |

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Stations

Summary: Retrieves the collection of station resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext



| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Stations/{stationCode}

Summary: Retrieves the specific station resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
stationCode	path	string	Yes	The unique station code.
cultureCode	query	string	No	The specific culture code.
eTag	query	string	No	The eTag.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/stations/{stationCode}/details

Summary: Get the station details.

Description

Note this endpoint is not cached like other resource endpoints.

Request Details

Parameters

Name	Location	Type	Required	Description
stationCode	path	string	Yes	The station code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/stations/category/{stationCategoryCode}

Summary: Gets a response with stations filtered by category.

Description

This endpoint is resource intensive until cached.

Request Details

Parameters

Name	Location	Type	Required	Description
stationCategoryCode	path	string	Yes	The unique station category code. Please note that this should be a char and not a string.

plaintext	

| ActiveOnly | query | boolean | No | Flag indicating to return just active only or all. || CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Suffixes

Summary: Retrieves the collection of suffix resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Suffixes/{suffixCode}

Summary: Retrieves the specific suffix resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
suffixCode	path	string	Yes	The unique suffix code.
cultureCode	query	string	No	The specific culture for the suffix.

Responses

404 Response

200 Response

GET /api/dcs/v1/resources/tailNumbers

Summary: Retrieves the tail number inventory.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The resource eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/TimeZones

Summary: Retrieves the collection of time zone resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/TimeZones/{timeZoneCode}

Summary: Retrieves the specific time zone resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
timeZoneCode	path	string	Yes	The unique time zone code.
cultureCode	query	string	No	The specific culture code.
eTag	query	string	No	The eTag.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/Titles

Summary: Retrieves the collection of title resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext 

|

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. || StartIndex | query | integer (int32) | No | The start index value for this request. || ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/Titles/{titleKey}

Summary: Retrieves the specific title resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
titleKey	path	string	Yes	The unique title key.
cultureCode	query	string	No	The specific culture for the title.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/TravelClasses

Summary: Retrieves the collection of travel class resources.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ActiveOnly	query	boolean	No	Flag indicating to return just active only or all.

plaintext	

| CultureCode | query | string | No | The desired culture code to be returned. || ETag | query | string | No | The cache token for this request. Data will not be returned if etag still matches. | | StartIndex | query | integer (int32) | No | The start index value for this request. | | ItemCount | query | integer (int32) | No | The item count value for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/TravelClasses/{travelClassCode}

Summary: Retrieves the specific travel class resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelClassCode	path	string	Yes	The unique travel class code.
cultureCode	query	string	No	The specific culture code.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/travelerNotifications

Summary: Retrieves a collection of traveler notification resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/resources/travelerNotifications/{notificationTypeCode}

Summary: Retrieves the specified traveler notification resource.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
notificationTypeCode	path	string	Yes	Traveler notification type.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/travelerNotifications/deliveryMethods

Summary: Retrieves the traveler notification delivery methods filtered by program code and level.

Description

This endpoint will return the allowed traveler notification delivery methods based on the program code and level. Any delivery methods that have a required program and level where the passed in values do not meet that level or above will be removed. If the customer is not enrolled in the default loyalty program, leave the program code and level null and the delivery methods will be filtered appropriately.

Request Details

Parameters

Name	Location	Type	Required	Description
programCode	query	string	No	The customer's default loyalty program code.
programLevel	query	string	No	The customer's program level of their default loyalty program.

Responses

404 Response

304 Response

200 Response

The list of traveler notification delivery methods that are allowed based on the customer's default loyalty program and program level.

GET /api/nsk/v1/resources/travelerNotifications/deliveryMethodSettings

Summary: Retrieves the notification delivery method settings resource.

Description

Contains information about the SMS, PUSH, and EMAIL notification delivery methods and their maximum number of allowed notifications.

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

The list of notification delivery method settings resource.

GET /api/nsk/v1/resources/zones/{zoneCode}/subZones

Summary: Retrieves the collection of subzones.

Description

This endpoint does not cache the resulting subzone resources.

Request Details

Parameters

Name	Location	Type	Required	Description
zoneCode	path	string	Yes	The unique code of the zone to which the subzones belong.

Responses

404 Response

200 Response

GET /api/nsk/v1/resources/zones/{zoneCode}/subZones/{subZoneCode}

Summary: Retrieves the specific subzone.

Description

This endpoint does not retrieve the subzone resource from the cache.

Request Details

Parameters

Name	Location	Type	Required	Description
zoneCode	path	string	Yes	The unique code of the zone to which the subzones belong.
subZoneCode	path	string	Yes	The unique subzone code.

Responses

404 Response

200 Response

Seatmaps Endpoints

GET /api/nsk/v4/seatmaps/{journeyKey}

Summary: Gets the list of seat maps for a specified journey.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The Journey Key.
IncludePropertyLookup	query	boolean	No	Flag indicating to include seat property lookup on the response.

plaintext



|

| CultureCode | query | string | No | The desired culture code. |

Responses

200 Response

The list of available seat maps.

400 Response

GET /api/nsk/v1/seatmaps/{legKey}

Summary: Gets the seat maps for a specified leg.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg Key.
IncludePropertyLookup	query	boolean	No	Flag indicating to include seat property lookup on the response.

plaintext 

| CultureCode | query | string | No | The desired culture code. |

Responses

200 Response

The seat map for the given leg key.

400 Response

Seats Endpoints

POST /api/dcs/v1/seats/block

Summary: Blocks seats on a specific piece of equipment.

Description

A blocked seat cannot be associated with/assigned to a passenger. The change can be verified on a unit's availability on a seat map.

Request Details

Request Body

The unit keys to block.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

DELETE /api/dcs/v1/seats/unblock

Summary: Unblocks seats on a specific piece of equipment.

Description

The change can be verified on a unit's availability on a seat map.

Request Details

Request Body

The unit keys to unblock.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Settings Endpoints

GET /api/nsk/v1/settings/booking

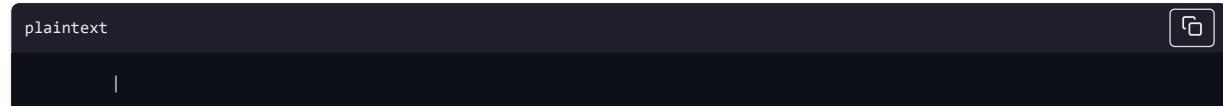
Summary: Get the general settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.



| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

401 Response

400 Response

304 Response

200 Response

Examples

Response 400

With Invalid Role Code:

```
json
{
  "errors": [
    {
      "id": "d05ebc4a-2534-428d-bdf0-284876cf8c64",
      "code": "nsk:Exceptions:OutOfRange",
      "message": "nsk:Exceptions:OutOfRange",
      "type": "Validation",
      "details": null,
      "rawMessage": "The value '1234' is out of range for 'SettingsQueryBase.RoleCode'.\nExpected values: (ABD1,ABD2,ABD3,ABD4,ABT
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}
```

Description: Example Error if role code is invalid

Response 200

Retrieve General Settings:

```

json
{
  "data": {
    "allowChangeSsrOnCheckedInPassenger": true,
    "allowBlockedSeatAssignment": false,
    "allowChangeOnRestrictedFlight": false,
    "allowPromoCodes": true,
    "allowEditableApiChannel": true,
    "allowEditableDirectChannel": true,
    "allowEditableGdsBookingWithTaPayment": true,
    "allowEditableGdsChannel": true,
    "allowEditableWebChannel": true,
    "allowGroupNameUpdates": true,
    "allowManifestRetrievalOnUnblockedFlightLines": true,
    "allowOverbookingSsrOnMove": false,
    "allowQueueingGdsBookingWithTaPopForETicketReissue": false,
    "allowRestrictedSeatAssignment": false,
    "allowSeatAssignment": 2,
    "bookingAccessControl": 7,
    "viewBookingsByControl": 3,
    "editBookingsByControl": 3,
    "defaultBookingComment": null,
    "doNotIncludeUnitProperties": true,
    "duplicateBookingDetectionTimeSpan": "00:00:00",
    "enableAutomaticSeatAssignment": true,
    "highlightFareOverride": 0,
    "highlightIrop": 0,
    "highlightScheduleChange": 0,
    "highlightStandby": 0,
    "highlightUpgrade": 0,
    "managePassiveSegments": true,
    "maskHistory": true,
    "reassignSeatingBehavior": 0,
    "restrictedExternalSystemCodes": [],
    "timeDisplay": 0,
    "allowSeatGroupRestrictionOverride": true,
    "allowAccessToComments": true,
    "sharedCommentTypes": {
      "reservation": true,
      "itinerary": true,
      "manifest": true,
      "alert": true
    },
    "allowOverrideOfSalesChannel": true,
    "autoFillContactInformationByPhoneNumber": false,
    "allowAccessToArchivedBooking": true,
    "allowQueueBookingsOnSpecifiedLeg": false,
    "queueFlightOpenTimeSpan": "00:00:00",
    "queueFlightCloseTimeSpan": "00:00:00"
  }
}

```

Description: Example response for successfully retrieving the booking general settings

GET /api/nsk/v1/settings/booking/checkin

Summary: Get the checkin settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/booking/contact

Summary: Get the contact settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/booking/customerAccount

Summary: Get the custom account settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/booking/fee

Summary: Get the fee settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext   
|
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/booking/flightSearch

Summary: Get the flight search settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext   
|
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/booking/general

Summary: Gets the non-role based general booking settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve General Booking Settings Response:

```
json 
```

```
{
  "data": {
    "maximumAgeForInfantInLap": 24,
    "useDirectCurrencyConversion": true,
    "clearInfantDetailsWhenInfantInLapIsDeleted": true,
    "mixedClassRouteFaresEnabled": true,
    "sumOfSectorFaresEnabled": true,
    "sumOfSectorProductClassPreCompression": true,
    "sellSeatSsrBeforeAssignment": true,
    "applyAllUpPricingGlobally": true,
    "repriceWhenCustomerProgramIsChanged": true,
    "recalculateHoldWithBookingChanges": true,
    "preventCommitBookingWithCancelledInventory": true,
    "defaultCarrierCode": "NV",
    "defaultBookingSubQueueCode": "Unknwn",
    "maximumNumberOfMarkets": 6,
    "interlineHoldBeforeDeparture": 2880,
    "warningTimeBeforeHoldCancel": 0,
    "interlineConnectionPartners": "G3",
    "applyFareForInLapInfantWithPassengerType": " ",
    "validateMaximumAgeBy": "FirstSegment",
    "validateMinimumAgeBy": "FirstSegment",
    "indiaGstRegistrationContactType": "None",
    "enforceStandardGroupBooking": true,
    "storeConfirmingWaivedFees": true,
    "personallyIdentifiableInformationMasking": true,
    "lowFareCacheMode": null,
    "retrieveFareInformationFromGeneralReferenceManagement": true,
    "storeDeletedManuallyAddedFees": true
  }
}
```

Description: Example response for successfully retrieving the general booking settings.

GET /api/nsk/v1/settings/booking/passenger

Summary: Get the passenger settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/settings/booking/payment

Summary: Get the role based booking payment settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve Booking Payment Settings v2:

```

json

{
  "data": {
    "allowExternalPaymentChargeBack": true,
    "allowHoldDateOverride": true,
    "allowIgnoreVoucherRestrictions": true,
    "allowVoucherIssuance": true,
    "allowManualAuthorization": true,
    "allowOverPay": true,
    "allowUnderPay": true,
    "allowUnvoidVoucher": true,
    "allowUseOfBinFraudTool": true,
    "allowVoidVoucher": true,
    "enableAutoPaymentVerification": true,
    "allowConfirmBookingWithPartialTicketing": true,
    "allowUnmaskingCreditCardNumber": true,
    "allowDisplayNoPaymentWarning": true,
    "requirePassengerOrContactForCreditFile": true,
    "requirePassengerOrContactForCreditShell": true,
    "requireCommentsForCreditShell": true,
    "disableAlternateAgAccount": true,
    "creditShellCreditCode": "BKGI30",
    "maxInstallmentOverride": 1,
    "maxPaymentQueries": 1,
    "secondsTillStartPaymentQuery": 1,
    "secondsTillRepeatPaymentQuery": 1,
    "holdsDaysBeforeDeparture": "10.00:00:00",
    "holdsDaysAfterDeparture": "-00:01:00",
    "allowPendingCustomerActionAuthorization": true,
    "allowManualAuthorizationOfPendingPaymentWithoutValidation": false,
    "maskPaymentFields": true
  }
}

```

Description: Example response for successfully retrieving the booking payment settings v2

GET /api/nsk/v1/settings/booking/reserveFlights

Summary: Get the reserve flight settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```

plaintext
| RoleCode | query | string | No | Gets or sets the role code for this request. |

```

Responses

404 Response

400 Response

304 Response

200 Response

Examples

Response 400

With Invalid Role Code:

Description: Example Error if role code is invalid

Response 200

Retrieve Booking Reserve Flights Settings:

```

json

{
  "data": {
    "availabilityDepartureTimeLimit": "-41.15:00:00",
    "availabilityMaxDaysOut": "60.00:00:00",
    "sellDepartureTimeLimit": "-416.15:00:00",
    "availableSpaceDisplayMax": 0,
    "flightTimeChangeToleranceMinutes": 0,
    "viewCapacity": false,
    "viewLidSold": false,
    "viewSsrAvailability": true,
    "viewManifestControl": 2,
    "adHocFlightControl": 0,
    "allowSellOverbook": 0,
    "allowMoveOverbook": 0,
    "allowUpgradeOverbook": 0,
    "allowSplitJourneyMoves": false,
    "allowMove": 2,
    "moveByMaximumClassAvailability": true,
    "requireSameCityPairOnMoves": false,
    "maxMoveDaysForward": "365.00:00:00",
    "maxMoveDaysBack": "365.00:00:00",
    "daysBeforeIrop": "365.00:00:00",
    "allowFareOverride": true,
    "allowClosedFlightFareOverride": true,
    "allowFareOverrideDuringDivide": true,
    "allowIgnoreRestrictions": true,
    "allowInvalidPriceStatus": true,
    "cancelDepartureTimeLimit": "-1000.00:00:00",
    "seatAssignmentTimeLimit": "-05:00:00",
    "allowUpgrade": false,
    "allowCreateAdhocConnect": true,
    "minimumTimeBetweenFlightsEnforcement": true,
    "upgradeDepartureTimeLimit": "01:00:00",
    "upgradeCabinStartOffset": "208.08:00:00",
    "minimumTimeBetweenFlights": "00:00:00",
    "modifyPassiveSegmentTimes": true,
    "allowSellOfFlightOverlap": false,
    "availabilityMaxDate": "9999-12-31T23:59:59.9999999",
    "allowedFareOverrideType": 2,
    "allowMoveOnInitialBooking": false,
    "allowDowngrade": true,
    "allowDowngradeOverbook": 3,
    "allowRestoreOverbook": 0,
    "enableCheckValidPrice": true,
    "enableTroubleshootingButton": true
  }
}

```

Description: Example response for successfully retrieving the booking reserve flights settings

GET /api/nsk/v1/settings/customerPrograms

Summary: Gets customer program settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/eTickets

Summary: Gets the non-role based e-ticket configuration.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/externalMessageControls

Summary: Gets the external message control settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve External Message Controls Settings:

```
json   
{  
  "data": {  
    "carrierCodes": [  
      "NV",  
      "AD",  
      "JQ",  
      "40"  
    ]  
  }  
}
```

Description: Example response for successfully retrieving the external message controls settings

GET /api/nsk/v1/settings/general/applicationLogon

Summary: Retrieves the logon settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext   
|
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/general/codes

Summary: Retrieves the codes settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext   
|
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/general/operations

Summary: Retrieves the operations settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext



|

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

400 Response

304 Response

200 Response

Examples

Response 400

With Invalid Role Code:

json



```
{  
  "errors": [  
    {  
      "id": "d05ebc4a-2534-428d-bdf0-284876cf8c64",  
      "code": "nsk:Exceptions:OutOfRange",  
      "message": "nsk:Exceptions:OutOfRange",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "The value '1234' is out of range for 'SettingsQueryBase.RoleCode'.\nExpected values: (ABD1,ABD2,ABD3,ABD4,ABT  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r  
    }  
  ],  
  "data": null  
}
```

Description: Example Error if role code is invalid

Response 200

Retrieve Operations Settings:

json



```
{
  "data": {
    "overrideOverWingExitRowRequirement": true,
    "iropFromDepartureDateOffset": "10.00:00:00",
    "allowDiversionIrop": true,
    "allowConsoleIrop": true,
    "allowCreateAdHocConnectionsInIrop": true,
    "allowIropOversellOnToFlight": true,
    "flightFollowingAccessControl": 7,
    "allowAircraftMishapUtilityAccess": true,
    "allowCheckedBaggageWeightChange": true,
    "allowOpeningOfClosedFlights": true,
    "timeBeforeDepartureThatCloseFlightIsAllowed": "10.00:00:00",
    "timeAfterDepartureThatOpenFlightIsAllowed": "10.00:00:00",
    "allowClearPassengerDataInAircraftMishap": true,
    "allowRestorePassengerDataInAircraftMishap": true,
    "allowInventoryStatusChange": true,
    "adHocFromDateOffset": "10.00:00:00",
    "bagDropMode": 0,
    "allowCrewBagAccess": true,
    "allowRushBagAccess": true,
    "allowCrewBagCheckInWhenFlightIsClosedPending": true,
    "allowRushBagCheckInWhenFlightIsClosedPending": true,
    "allowMarketingCheckIn": true,
    "verifyAPISInternationalFlight": true,
    "allowInventorySSRMaintenanceUpdate": true,
    "allowInventoryRecordMaintenanceUpdate": true,
    "allowCancelRestoreInventory": true,
    "allowDiversionAdHoc": true,
    "allowManageTailNumber": true,
    "allowEquipmentSwapUtilityAccess": true,
    "allowRZAccess": true,
    "allowUpdateStandbyPriorityCode": true,
    "allowManageVoucher": true,
    "flightInformationPermissions": 1,
    "allowSegmentsByTime": true,
    "allowBaggageTrackingAccess": true,
    "enableFlightSearchRestrictionByOrigin": true,
    "allowBaggageModeSwitchToIBCMode": true,
    "daysForwardAllowedForSearchingCheckInAndReports": "30.00:00:00",
    "daysPriorAllowedForSearchingCheckInAndReports": "30.00:00:00",
    "sendAPISReport": true,
    "watchList": {
      "allowViewForWatchList": true,
      "allowDetailViewForWatchList": true,
      "allowClearPassengerFromWatchList": true
    },
    "programs": {
      "sendSITAMessage": true,
      "sendMessage": true,
      "adHocScheduleInventoryConnections": true,
      "allowCAPPSSReportAccess": true,
      "allowCheckInSelectee": true
    },
    "reports": {
      "enplanementDeplanement": true,
      "manifestConnectionInfo": true,
      "manifestDetailedInfo": true,
      "flightLoad": true,
      "checkIn": true,
      "aircraftZone": true,
      "seatAssignment": true,
      "flightClose": true,
      "paymentReceipt": true,
      "paymentReceiptConfigurations": {
        "displayLocationPrompt": true
      },
      "checkedBaggage": true,
      "availability": true,
      "bagWeightByLeg": true,
      "selecteesByFlight": true,
      "allowCheckIn": true
    }
  }
}
```

```

        "mealFulfillment": true,
        "flightManifest": true,
        "flightManifestConfigurations": {
            "identifySelecteePassengers": true,
            "identifyPassengersWithBalanceDue": true
        },
        "prePostFlightManifest": true,
        "cabinServiceArea": true,
        "displayPNRInCabinServiceAreaAndMealFulfillmentReports": true
    }
}
}
}

```

Description: Example response for successfully retrieving the operations settings

GET /api/nsk/v1/settings/general/organization

Summary: Gets the organization settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/general/session

Summary: Retrieves the session settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext 

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/itinerary

Summary: Gets the itinerary settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/loyalty

Summary: Gets the non-role based loyalty settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/notifications/general

Summary: Gets the general New Skies notification settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve Notifications Settings:

json

```
{  
  "data": {  
    "enableTicketControlNumberNotifications": true  
  }  
}
```



Description: Example response for successfully retrieving the notifications settings

GET /api/nsk/v1/settings/payment

Summary: Get the non role based payment settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/phoneNumberValidation

Summary: Gets the non-role based phone number validation settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/settings/premiumServices

Summary: Retrieves the premium services settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve Premium Services Settings:

```
json   
{  
  "data": {  
    "governmentDataProtectionRequirements": true,  
    "accessToExternalBaggageSystems": true,  
    "unitedKingdomInteractiveAdvancePassengerProcessing": true,  
    "advancedParentChildTravelAgencyManagement": true,  
    "doNotPersistExternalPaymentDetails": false,  
    "baggageSelfTagging": true,  
    "loyaltyCustomerAccountView": true,  
    "pricingOptimizationServices": true,  
    "secureFlight": true,  
    "allInclusivePricingDisplay": true,  
    "numericAddressVerificationService": true,  
    "koreanInteractiveAdvancePassengerProcessing": true,  
    "taiwanInteractiveAdvancePassengerProcessing": true,  
    "canadaInteractiveAdvancePassengerProcessing": true,  
    "attributeShopping": true,  
    "documentCheckConfiguration": true,  
    "transmitFareDataToAirlineTariffPublishingCompany": true,  
    "travelNotifications": true,  
    "bundles": true  
  }  
}
```

Description: Example response for successfully retrieving the premium services settings

GET /api/nsk/v1/settings/serviceBundles

Summary: Gets the service bundles settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext   
|  
|
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v2/settings/skySpeed

Summary: Retrieves the role-based and non-role based SkySpeed settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

plaintext	

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

304 Response

200 Response

Examples

Response 200

Retrieve SkySpeed Settings v2:

```
json

{
  "data": {
    "disableBaseCurrency": false,
    "manifestViewDate": 0,
    "disableOutboundTime": true,
    "maximumNumberOfTtyMessagesPerBatchForDeletion": 1000,
    "excludeOutboundTimeTab": true,
    "postalCodeLookup": 0,
    "disableReturnTime": true,
    "serviceServerName": 0,
    "excludeReturnTimeTab": true,
    "enableViewPaymentAllocation": true,
    "enableShowValidFaresCheckbox": true,
    "fareBasisInRegularBooking": 9,
    "fareBasisInOverbook": 9,
    "displayReservedFlights": true,
    "displaySsrNestValues": true,
    "defaultFocusToCustomerIdField": true,
    "allowDeleteComments": true,
    "promptCreateInternetUserId": true,
    "allowManagePasswordForCustomerAccount": true,
    "viewFareQuote": true,
    "allowNonCarrierOrgToUseOtherAgency": true,
    "viewTeletypeProcessing": true,
    "viewGdsHistoryDialog": true,
    "returnDateDefaultToOutboundDate": true,
    "displayFyiNotesOnLogin": true,
    "displayResidentCountry": true,
    "displayPricesByPassengerType": null,
    "defaultToCustomerCreditCard": true,
    "forceViewMandatoryComments": true,
    "showTravelDocumentsOnPassengersScreen": true,
    "showDestinationAddressesOnPassengersScreen": true,
    "showFrequentFlyerNumberOnPassengersScreen": true,
    "showMealSsrButtonOnPassengersScreen": true,
    "allowETicketing": true,
    "allowFeesToDelete": true,
    "displayReferenceNumber": true,
    "enableLoyaltyCustomerAccountView": true,
    "enableLoyaltyFareDisplay": true,
    "displayETicketRecords": true,
    "allowETicketCouponStatusToUpdate": true,
    "enableRepriceButtonOnReservedFlightsScreen": true,
    "enableSplitCancelButtonOnReservedFlightsScreen": true,
    "allowTravelNotificationsAccess": true,
    "allowThemeEditorAccess": true,
    "allowBlacklistAccessAndRetrieveReservationsFromPayments": true,
    "maskBookingContactInformation": false,
    "accessToFlightsWithTrafficRestrictions": true,
    "enabledTicketCodes": null,
    "reservationComments": null,
    "itineraryComments": null,
    "manifestComments": null,
    "enableDeletePassengerButton": true,
    "showMultiCityShoppingTabOnAvailabilityRequestScreen": true
  }
}
```

Description: Example response for successfully retrieving the SkySpeed settings v2

GET /api/nsk/v1/settings/system/general

Summary: Gets the role-based general system settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
ETag	query	string	No	The cache eTag for this request.

```
plaintext 
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

The item was not found.

304 Response

The requested item has not been modified.

200 Response

OK. Request completed successfully.

Examples

Response 404

Settings not found.:

```
json   
null
```

Description: Example when the settings cannot be retrieved.

Response 304

Matching ETag:

```
json   
null
```

Description: Response when an ETag is provided and the cached data still matches.

Response 200

Settings are retrieved successfully:

```
json   
{  
  "data": {  
    "manageFYIContent": 7,  
    "manageGeneralReferenceManagement": 7  
  }  
}
```

Description: Example when the settings are retrieved successfully.

GET /api/nsk/v1/settings/systemConfiguration/finance

Summary: Retrieves the session system configuration finance setting.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The cache eTag for this request.

```
plaintext
```

| RoleCode | query | string | No | Gets or sets the role code for this request. |

Responses

404 Response

200 Response

Examples

Response 200

Retrieve System Configuration Finance:

```
json
```

```
{
  "data": {
    "financeAccess": 7,
    "financeData": {
      "manageAccounts": 2,
      "manageCreditTypes": 6,
      "manageCurrency": 4,
      "manageFareDiscountCodes": 0,
      "manageFareSurcharges": 1,
      "managePaymentTypes": 5,
      "managePromotionCodes": 7,
      "manageTaxesAndFees": 1,
      "allowAgentToImportFeeVariations": true,
      "manageVouchers": 1,
      "allowAccountReconciliation": true,
      "allowBalanceAndCreditLimitUpdate": true,
      "allowAccountStatementAdjustment": true
    },
    "manageFinancialControlLockDate": 0
  }
}
```

Description: Example response for successfully retrieving the system configuration - finance settings

GET /api/nsk/v1/settings/travelerNotification

Summary: Gets the traveler notification settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The eTag.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/user/agencyCreation

Summary: Retrieves the agency creation settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

GET /api/nsk/v1/settings/user/customerCreation

Summary: Retrieves the customer creation settings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
eTag	query	string	No	The unique etag ID.

Responses

404 Response

304 Response

200 Response

Token Endpoints

PUT /api/auth/v1-beta/token

Summary: BETA - Updates the short lived JWT.

Description

This is a beta endpoint and is not currently a long term supported endpoint. This endpoint may be removed or changed in future updates.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

401 Response

GET /api/nsk/v1/token

Summary: Get the information about the current token.

Description

With custom application names (system type in NewSkies), because of the nature of the underlying system type, the `NskServer.SystemName` will always show as `UserDefined`. To see the custom name, see the `ServerContext.ApplicationName`.

Request Details

Responses

200 Response

The generated token.

DELETE /api/nsk/v1/token

Summary: Abandons the active token.

Description

No description available

Request Details

Responses

200 Response

PUT /api/nsk/v2/token

Summary: Given a non-null request, upgrades the current session's logged in user. Otherwise, keeps the active token alive.

Description

Security Note: This endpoint has the ability to log in as any role assigned to a user or that is allowed through impersonation settings. A user may still log into a role that is not assigned to them if the impersonation settings allow it.

Request Details

Request Body

Content Type: `application/json`

Schema: See API specification for detailed schema

Responses

200 Response

201 Response

POST /api/auth/v1-beta/token/anonymous

Summary: BETA - Create a short lived anonymous JWT.

Description

This is a beta endpoint and is not currently a long term supported endpoint. This endpoint may be removed or changed in future updates. This endpoint works differently than the v2 token. Tokens will now have a set expiration time. Client applications should call the put endpoint to get a new token before this expiration time.

Request Details

Request Body

Get token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v1/token/culture

Summary: Sets the culture code for the lifetime of the token.

Description

Note there is middleware that looks for "Culture" on every request in the headers or query string parameters. These can be used if you want to set the culture for a single request only.

Request Details

Parameters

Name	Location	Type	Required	Description
defaultCultureCode	query	string	No	No description

Responses

200 Response

POST /api/auth/v1/token/multifactor

Summary: Register multi-factor authentication options.

Description

Allows first time users to select their multifactor authentication method and will send a challenge code to that method.

Request Details

Request Body

The multi-factor registration request

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

202 Response

Waiting on additional data.

400 Response

Bad request.

500 Response

Error.

Examples

Request

Register Email:

```
json
{
  "credentials": {
    "domain": "DEF",
    "username": "agent123",
    "password": "*****"
  },
  "registration": {
    "email": "getChallengeCode@mfa.com",
    "type": 0
  }
}
```

Description: Initial request to register an email.

Verify:

```
json
{
  "credentials": {
    "domain": "DEF",
    "username": "agent123",
    "password": "*****"
  },
  "verify": {
    "challengeCode": "012345",
    "challengeId": "01234567890123456789"
  }
}
```

Description: Second request to verify the challenge code with a challenge id.

Register SMS:

```
json
{
  "credentials": {
    "domain": "DEF",
    "username": "agent123",
    "password": "*****"
  },
  "registration": {
    "phone": "123-123-1234",
    "type": 1
  }
}
```

Description: Initial request to register an phone number.

Register TOTP:

```
json
{
  "credentials": {
    "domain": "DEF",
    "username": "agent123",
    "password": "*****"
  },
  "registration": {
    "type": 2
  }
}
```

Description: Initial request to register an authenticator app.

Response 201

Verified Example:

```
json   
{  
  "data": {  
    "qrCodeUrl": null,  
    "challengeId": null  
  }  
}
```

Description: MFA registrations has been verified. This will only need to be done once and the user can now proceed to POST /token/user to restart the authentication process.

Response 202

TOTP Example:

```
json   
{  
  "data": {  
    "qrCodeUrl": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAeAAAQAAQABgAABgAAQAAADjFjCXAAADHE1EQVR42u2dW47bMMaxFiXoBxpK27iV5AQY0jsyXkk  
    "challengeId": "01234567890123456789"  
  }  
}
```

Description: This value can be used in websites or by devs using sites such as <https://jaredwinick.github.io/base64-image-viewer/> and would be scanned by an authenticator app to provide the challenge code. Next please call this same endpoint to verify the MFA registration.

Email or SMS Example:

```
json   
{  
  "data": {  
    "qrCodeUrl": null,  
    "challengeId": "01234567890123456789"  
  }  
}
```

Description: An email will be sent that provides a challenge code. Next please call this same endpoint to verify the MFA registration. If registration is not verified then it will timeout after 10 minutes and will be required again in future authentication calls.

Response 400

Already registered.:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk-server:MFA:Registered",  
      "message": "Already registered or waiting for the registration challenge code. To restart the registration process try again.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: This can occur when already registered or waiting for the registration challenge code. Please try again later.

Response 500

MFA service down.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Credentials:Issue",
      "message": "nsk-server:Credentials:Issue",
      "type": "Error",
      "details": null,
      "rawMessage": "Error calling IMFAService.authenticateAdmin: MFA service unavailable.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Internal server problem with authentication. Please try again later.

PUT /api/auth/v1-beta/token/role

Summary: BETA - Updates the short lived JWT with new role permissions.

Description

This is a beta endpoint and is not currently a long term supported endpoint. This endpoint may be removed in future updates.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

401 Response

POST /api/nsk/v1/token/serverTransfer

Summary: Transfers the server context from NewSkies and returns a new token.

Description

This endpoint returns a new token. The new token must be applied in the authorization header to access the transferred server context.

Request Details

Request Body

The server transfer request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The generated token for the transferred server context.

POST /api/nsk/v1/token/singleSignOn

Summary: Creates the general access token that will grant access to the API.

Description

In order to use single sign on, a single sign on token must first be associated with a user. Refer to the "user/singleSignOnToken" endpoints.

Request Details

Request Body

The single sign on credentials.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The generated token.

PUT /api/nsk/v1/token/singleSignOn

Summary: Given a non-null request, upgrades the current session's logged in user. Otherwise, keeps the active token alive.

Description

In order to use single sign on, a single sign on token must first be associated with a user. Refer to the "user/singleSignOnToken" endpoints.

Request Details

Request Body

The single sign on credentials.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

201 Response

POST /api/auth/v1-beta/token/user

Summary: BETA - Creates a short lived JWT specific to a user.

Description

This is a beta endpoint and is not currently a long term supported endpoint. This endpoint may be removed or changed in future updates. This endpoint works differently than the v2 token. Tokens will now have a set expiration time. Client applications should call the put endpoint to get a new token before this expiration time.

Request Details

Request Body

The nsk token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

202 Response

Waiting on additional data.

400 Response

Bad request.

500 Response

Error.

Examples

Request

Standard Request:

```
json   
{  
  "credentials": {  
    "domain": "DEF",  
    "username": "agent123",  
    "password": "*****"  
  }  
}
```

Description: Simple request.

Multifactor Request:

```
json   
{  
  "credentials": {  
    "domain": "DEF",  
    "username": "agent123",  
    "password": "*****"  
  },  
  "multiFactor": {  
    "challengeCode": "123456",  
    "challengeId": "01234567890123456789"  
  }  
}
```

Description: Request after receiving multifactor challenge code.

Response 201

Success.:

```
json   
{  
  "data": {  
    "multiFactorRequired": null,  
    "token": "**JWT Token**",  
    "cultureCode": "en-US",  
    "currencyCode": "USD",  
    "roleCode": "AIR2",  
    "locationCode": "HQ",  
    "domainCode": "DEF",  
    "organizationCode": "88888888",  
    "expires": "2025-08-07T05:02:16.9603017+00:00"  
  }  
}
```

Description: Login success.

Response 202

MFA registration required.:

```
json   
{  
  "messages": [  
    {  
      "code": "nsk-server:MFA-RegistrationRequired",  
      "type": "Information",  
      "value": "Multi-factor registration is required.",  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": {  
    "multiFactorRequired": {  
      "challengeId": null,  
      "requiresRegistration": true,  
      "requiresChallengeCode": false,  
      "types": [  
        2,  
        1,  
        0  
      ]  
    },  
    "token": null,  
    "cultureCode": null,  
    "currencyCode": null,  
    "roleCode": null,  
    "locationCode": null,  
    "domainCode": null,  
    "organizationCode": null,  
    "expires": null  
  }  
}
```

Description: This role requires multi-factor authentication. This user has not yet chosen the preferred type of MFA and registered with it. See POST /token/multifactor

MFA challenge required.:

```
json   
{  
  "messages": [  
    {  
      "code": "nsk-server:MFA-ChallengeCodeRequired",  
      "type": "Information",  
      "value": "Multi-factor authentication required. Please check your second factor method and enter the challenge code.",  
      "status": 0,  
      "details": null  
    }  
  ],  
  "data": {  
    "multiFactorRequired": {  
      "challengeId": "01234567890123456789",  
      "requiresRegistration": false,  
      "requiresChallengeCode": true,  
      "types": [  
        0  
      ]  
    },  
    "token": null,  
    "cultureCode": null,  
    "currencyCode": null,  
    "roleCode": null,  
    "locationCode": null,  
    "domainCode": null,  
    "organizationCode": null,  
    "expires": null  
  }  
}
```

Description: This role requires multi-factor authentication. This user has already registered but needs to use their chosen registration method for the 2nd factor authentication.

Response 400

Challenge code failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Credentials:Failed",
      "message": "nsk-server:Credentials:Failed",
      "type": "Error",
      "details": null,
      "rawMessage": "Challenge code failed validation for user DEF/authAppMFAUser.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: For security unless in debug mode for testing servers the same error will be returned for multiple credential scenarios.

Agent not found:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Credentials:Failed",
      "message": "nsk-server:Credentials:Failed",
      "type": "Error",
      "details": null,
      "rawMessage": "No agent found for requested agent name DEF/authAppMFAUser.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: For security unless in debug mode for testing servers the same error will be returned for multiple credential scenarios.

Invalid password:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Credentials:Failed",
      "message": "nsk-server:Credentials:Failed",
      "type": "Error",
      "details": null,
      "rawMessage": "The agent (DEF/authAppMFAUser) failed authentication.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: For security unless in debug mode for testing servers the same error will be returned for multiple credential scenarios.

Response 500

MFA service down.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Credentials:Issue",
      "message": "nsk-server:Credentials:Issue",
      "type": "Error",
      "details": null,
      "rawMessage": "Error calling IMFAService.authenticateAdmin: MFA service unavailable.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Internal server problem with authentication. Please try again later.

PUT /api/auth/v1-beta/token/user

Summary: BETA - Updates an anonymous short lived JWT to login as a user.

Description

This is a beta endpoint and is not currently a long term supported endpoint. This endpoint may be removed or changed in future updates. This endpoint works differently than the v2 token. Tokens will now have a set expiration time. Client applications should call the put endpoint to get a new token before this expiration time.

Request Details

Request Body

The nsk token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

202 Response

Waiting on additional data.

Examples

Response 202

TOTP Example:

```
json

{
  "data": {
    "qrCodeUrl": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAeAAAQAAAQAAAADjFjCXAAADHE1EQVR42u2dw47bMAXFiXoBXpK27iV5AQY0jsyXkk",
    "challengeId": "01234567890123456789"
  }
}
```

Description: This value can be used in websites or by devs using sites such as <https://jaredwinick.github.io/base64-image-viewer/> and would be scanned by an authenticator app to provide the challenge code. Next please call this same endpoint to verify the MFA registration.

Email or SMS Example:

```
json  
  
{  
  "data": {  
    "qrCodeUrl": null,  
    "challengeId": "01234567890123456789"  
  }  
}
```

Description: An email will be sent that provides a challenge code. Next please call this same endpoint to verify the MFA registration. If registration is not verified then it will timeout after 10 minutes and will be required again in future authentication calls.

POST /api/nsk/v1/token/user/person/singleSignOn

Summary: Creates a new user, links it with a provider, then logs in as the new user.

Description

Warning: This endpoint currently only works with Facebook.

```
plaintext  
  
If this endpoint fails it could be because an SSO token is already  
linked to a user or the SSO token is invalid.
```

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The generated token.

Trip Endpoints

POST /api/nsk/v5/trip

Summary: Creates a new booking with the provided journeys, contacts, passengers, and SSRs.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel documents' gender. This will update the gender of all the travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.

plaintext



|

Request Body

The booking sell request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

400 Response

Examples

Response 400

Passenger type not allowed for infant in lap.:

json



```
{  
  "errors": [  
    {  
      "id": "51e618c6-8bdc-4ae1-8786-273382c58442",  
      "code": "nsk:Passenger:PassengerTypeNotAllowedForInfantInLap",  
      "message": "Passenger with type CHD is not allowed to have an infant in lap.",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "Passenger with type CHD is not allowed to have an infant in lap.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example when attempting to add an infant in lap to a passenger type that is not allowed.

No POST request body.:

json



```
{  
  "errors": [  
    {  
      "id": "f1492787-eeb5-4ead-bc82-35e8ceddb7a0",  
      "code": "nsk:Exceptions:Null",  
      "message": "An unexpected null value of type 'BookingSellRequest'.",  
      "type": "Validation",  
      "details": null,  
      "rawMessage": "An unexpected null value of type 'BookingSellRequest'.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example when making a call without a request body.

POST /api/nsk/v2/trip/info

Summary: Gets the search results for the trip information, based on the search criteria.

Description

This endpoint will return a maximum of 100 journeys. If more than 100 journeys are needed, the search criteria will need to be modified and additional trip calls will need to be made.

Request Details

Request Body

The trip information request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

A trip information response.

404 Response

GET /api/nsk/v1/trip/info/{legKey}/details

Summary: Retrieves cabin and various details for the specified leg key.

Description

The additional trip details include the leg cross references, the cabin details (nest counts), and the checkin-lid.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The encoded leg key.

Responses

200 Response

404 Response

PATCH /api/nsk/v1/trip/info/{legKey}/status

Summary: Stateless operation for updating the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes. An empty string or whitespaces are allowed input values and are not considered as null inputs. Null values will retain the original value in the trip status while empty strings or whitespaces will clear the value from the trip status. When updating tail numbers, this endpoint will not validate mismatching equipment types. If validation is required for tail numbers, please see `POST /equipment/tailNumber`.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.

Request Body

The trip information status request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

500 Response

Examples

Response 200

Update flight following information successful:

```
json   
{  
    "data": null  
}
```

Description: Example response when flight following information is updated successfully.

Mismatch tail number equipment type:

```
json   
{  
    "messages": [  
        {  
            "code": "nsk:TripStatus:EquipmentTypeMismatch",  
            "type": "Information",  
            "value": "nsk:TripStatus:EquipmentTypeMismatch",  
            "status": 0,  
            "details": {  
                "flightEquipmentType": "767",  
                "tailNumberEquipmentType": "731"  
            }  
        }  
    ],  
    "data": null  
}
```

Description: Example response when the tail number is updated even if the equipment type is mismatching.

Response 400

Update Flight Following Information Not Allowed:

```
json   
{  
    "errors": [  
        {  
            "id": "7a9504cd-8764-84af-9769-d634a09cd09d",  
            "code": "nsk-server:AuthorizationManageFLIFONotAllowed",  
            "message": "nsk-server:AuthorizationManageFLIFONotAllowed",  
            "type": "Validation",  
            "details": {  
                "message": "Agent does not have the proper clearance to update FLIFO."  
            },  
            "rawMessage": "Agent does not have the proper clearance to update FLIFO.",  
            "exception": null,  
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
        }  
    ],  
    "data": null  
}
```

Description: Example response when an agent is not allowed to update the flight following information.

Leg key is invalid:

```
json
{
  "errors": [
    {
      "id": "c2f65a8c-e8e9-26a8-302d-7b3531bdc569",
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'legKey' with value 'NjM4NjY2ODkyMDAwMTMhICFTTEMhREVOITIzNTk3OTg-' is invalid.",
      "type": "Validation",
      "details": null,
      "rawMessage": "The identifier 'legKey' with value 'NjM4NjY2ODkyMDAwMTMhICFTTEMhREVOITIzNTk3OTg-' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the leg key is invalid.

Tail number does not exist:

```
json
{
  "errors": [
    {
      "id": "8577fba3-d2e9-b0ff-c870-2e578bccb09b",
      "code": "nsk:TripStatus:TailNumberDoesNotExist",
      "message": "nsk:TripStatus:TailNumberDoesNotExist",
      "type": "Validation",
      "details": {
        "tailNumber": "abc"
      },
      "rawMessage": "The tail number does not exist.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the tail number does not exist. See /resources/tailNumbers for the list of all available tail numbers.

Tail number exceeds 10 characters:

```
json
{
  "errors": [
    {
      "id": "5a82f7d2-f6ce-01dd-73b4-49a1329a4c44",
      "code": "TripInformationStatusRequestBase.TailNumber:StringLengthAttribute",
      "message": "The field TripInformationStatusRequestBase must be a string with a maximum length of 10.",
      "type": "Validation",
      "details": {
        "validation": "StringLengthAttribute",
        "model": "TripInformationStatusRequestBase",
        "member": "TailNumber"
      },
      "rawMessage": "The field TripInformationStatusRequestBase must be a string with a maximum length of 10.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the tail number exceeds 10 characters.

Request is null:

```
json

{
  "errors": [
    {
      "id": "92f48a26-3bbc-b591-8712-34ddbe2ab6e5",
      "code": "core:Validation:NullRequest",
      "message": "The request is null or couldn't be converted to the expected data type.",
      "type": "Error",
      "details": null,
      "rawMessage": "The request is null or couldn't be converted to the expected data type.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    ],
    "data": null
  }
}
```

Description: Example response when the request is null.

GET /api/nsk/v2/trip/info/{legKey}/status

Summary: Retrieves the trip status data for the specified leg key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The encoded leg key.

Responses

200 Response

OK. Request completed successfully.

404 Response

The item was not found.

Examples

Response 200

Retrieve Trip Status:

json



```
{
  "data": {
    "tripStatusKey": "MzAxMzx2NxRxNCI0MTAxOCB0VjU2MDggU0xDQk8J",
    "operationDetails": {
      "tailNumber": null,
      "equipmentType": "380",
      "equipmentTypeSuffix": "800",
      "status": 0,
      "identifier": {
        "identifier": "5608",
        "carrierCode": "NV",
        "opSuffix": null
      },
      "tripOperationTimes": null,
      "counts": {
        "boardingCount": 0,
        "boardingSequence": 0,
        "infantCount": 0,
        "nonRevenueCount": 0,
        "revenueCount": 0,
        "checkInCount": 0,
        "fuelOnboardOut": 0,
        "fuelOnboardIn": 0,
        "upLiftFuel": 0,
        "upLiftWeight": 0,
        "cockpitCrewCount": 0,
        "cabinCrewCount": 0
      },
      "delays": [
        {
          "delayCode": "TEST",
          "delayMinutes": 10,
          "tripOperationDelayKey": "MzAxMzc1MyExMzU1",
          "note": "TEST DELAY EDIT",
          "createdDate": "2025-08-07T04:47:17.0396192Z",
          "includeMvtMessage": true
        }
      ],
      "payloads": null,
      "crewList": null,
      "comments": [
        {
          "text": "TEST COMMENT EDIT",
          "tripOperationCommentKey": "MzAxMzc1MyEzMDU4",
          "type": 0
        }
      ],
      "historyList": [
        {
          "tripOperationHistoryKey": "MzAxMzc1MyEyODkxNlsz",
          "tripOperationHistoryCode": "GP",
          "detail": "US|iAPI Scheduled Msg ( 0 / 0)",
          "createdHistoryDate": "2025-08-07T04:47:17.0396233+00:00"
        },
        {
          "tripOperationHistoryKey": "MzAxMzc1EyS1OBk2MsN3",
          "tripOperationHistoryCode": "GS",
          "detail": "US|iAPI Scheduled Msg ( 2 / 1)",
          "createdHistoryDate": "2025-08-07T04:47:17.0396245+00:00"
        }
      ]
    },
    "originDetails": {
      "status": 0,
      "note": null,
      "gate": null,
      "station": "SLC"
    },
    "destinationDetails": {
      "status": 0,
      "note": null,
      "gate": null,
      "station": "HNL"
    }
  }
}
```

```

        "baggageClaim": null,
        "gate": null,
        "station": "BOS"
    },
    "modifiedAgentName": "ExampleAgent",
    "verified": true,
    "irop": true,
    "modifiedDate": "2025-08-07T04:47:17.0396266Z"
}
}

```

Description: Successfully retrieving the trip status.

POST /api/nsk/v1/trip/info/{legKey}/status/comments

Summary: Stateless operation for creating operation comments in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.

Request Body

The trip operation comment create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```

json
{
  "errors": [
    {
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",
      "code": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "message": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "type": "Validation",
      "details": {
        "message": "Agent does not have the proper clearance to update FLIFO."
      },
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned." }
  ],
  "data": null
}

```

Description: Example response when an agent is not allowed to update the flight following information.

PUT /api/nsk/v1/trip/info/{legKey}/status/comments/{commentKey}

Summary: Stateless operation for updating the operation comments in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.
commentKey	path	string	Yes	The key of the comment to be updated.

Request Body

The trip operation comment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

500 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```
json Copy

{
  "errors": [
    {
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",
      "code": "nsk-server:AuthorizationManagerFLIFONotAllowed",
      "message": "nsk-server:AuthorizationManagerFLIFONotAllowed",
      "type": "Validation",
      "details": {
        "message": "Agent does not have the proper clearance to update FLIFO."
      },
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  

    }
  ],
  "data": null
}
```

Description: Example response when an agent is not allowed to update the flight following information.

DELETE /api/nsk/v1/trip/info/{legKey}/status/comments/{commentKey}

Summary: Stateless operation for deleting the operation comments in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.
commentKey	path	string	Yes	The key of the comment to be deleted.

Responses

200 Response

400 Response

500 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```
json   
{  
  "errors": [  
    {  
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",  
      "code": "nsk-server:AuthorizationManageFLIFONotAllowed",  
      "message": "nsk-server:AuthorizationManageFLIFONotAllowed",  
      "type": "Validation",  
      "details": {  
        "message": "Agent does not have the proper clearance to update FLIFO."  
      },  
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when an agent is not allowed to update the flight following information.

POST /api/nsk/v1/trip/info/{legKey}/status/delays

Summary: Stateless operation for creating operation delays in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.

Request Body

The trip operation delay request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",  
      "code": "nsk-server:AuthorizationManageFLIFONotAllowed",  
      "message": "nsk-server:AuthorizationManageFLIFONotAllowed",  
      "type": "Validation",  
      "details": {  
        "message": "Agent does not have the proper clearance to update FLIFO."  
      },  
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when an agent is not allowed to update the flight following information.

PATCH /api/nsk/v1/trip/info/{legKey}/status/delays/{delayKey}

Summary: Stateless operation for patching the operation delays in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.
delayKey	path	string	Yes	The key of the delay to be patched.

Request Body

The trip operation delay request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

500 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```

json

{
  "errors": [
    {
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",
      "code": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "message": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "type": "Validation",
      "details": {
        "message": "Agent does not have the proper clearance to update FLIFO."
      },
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when an agent is not allowed to update the flight following information.

DELETE /api/nsk/v1/trip/info/{legKey}/status/delays/{delayKey}

Summary: Stateless operation for deleting trip operation delays in the trip information status for a specific flight.

Description

A flight must be within three days of leaving to be able to make flight following information changes.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key of the flight being updated.
delayKey	path	string	Yes	The key of the delay to be deleted.

Responses

200 Response

400 Response

500 Response

Examples

Response 400

Update Flight Following Information Not Allowed:

```

json

{
  "errors": [
    {
      "id": "7a9504cd-8764-84af-9769-d634a09cd09d",
      "code": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "message": "nsk-server:AuthorizationManageFLIFONotAllowed",
      "type": "Validation",
      "details": {
        "message": "Agent does not have the proper clearance to update FLIFO."
      },
      "rawMessage": "Agent does not have the proper clearance to update FLIFO.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when an agent is not allowed to update the flight following information.

PUT /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/cancel

Summary: Updates the leg status to cancel.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

400 Response

PUT /api/nsk/v2/trip/info/{legKey}/status/operationDetails/status/closeLeg

Summary: Updates a leg status closed.

Description

A leg status can be set to closed regardless of check-in configuration period of time set in settings. This check-in restriction determines if a flight can be closed before a designated departure time. It is recommended to set a leg status to closed using the check-in restrictions. There are a number of factors that can prevent the leg from being closed. These include:

```

plaintext

A PNRGOV message is required and has not been sent
A flight closure or APIS message is required and has not been sent
Passengers have a balance due (unless overrideBalanceDueWarning is specified)

Passengers have a balance due and not all passengers on the booking
have the same lift status.

Exit Row Requirements are not met (unless OverrideExitRowRequirement is specified)

```

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Request Body

Flight close options.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

PUT /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/closePendingLeg

Summary: Updates a leg status to closed pending.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
overrideExitRowRequirement	query	boolean	No	This will allow the leg to be set to closed pending regardless of any exit row requirements.

plaintext
copy

Responses

200 Response

400 Response

PUT /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/mishap

Summary: Secures all passenger information associated with the flight leg clearing and locking the passenger data. This changes the leg status to Mishap.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

400 Response

401 Response

Examples

Response 400

Invalid Flight Key:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "The identifier 'legKey' with value 'aPQxScmtKWMS7YtIi6Pdp9kwUQKLrqc6j9K5f4cM' is invalid.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example of invalid flight key.

Update Failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:UpdateFailed",
      "message": "Attempted update of object 'legKey' has failed. Please verify this object exists.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example of when attempted status change to 'Mishap' failed.

PUT /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/openLeg

Summary: Updates a leg status to open.

Description

A leg status can be set to open regardless of checkin configuration period of time set in settings. This checkin restriction determines if a flight can be opened after a designated departure time. It is recommended to set a leg status to open using the checkin restrictions.

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.
bypassCheckinRestrictions	query	boolean	No	This will allow updates regardless of checkin configuration period of time set in settings.

plaintext	
convertNoShowToDefault query boolean No If true, converts no show passengers' liftStatus to default	

Responses

200 Response

400 Response

PUT /api/nsk/v1/trip/info/{legKey}/status/operationDetails/status/restore

Summary: Updates the leg status to normal.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
legKey	path	string	Yes	The leg key.

Responses

200 Response

400 Response

POST /api/nsk/v2/trip/info/legs

Summary: Searches for the inventory legs based on the search criteria.

Description

This is functionally a GET but POST is used to reduce URL length overhead.

Request Details

Request Body

The trip information leg query.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v2/trip/info/legs/simple

Summary: Performs a simplified search for trip information by leg, based upon minimal, required data for the search criteria.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Origin	query	string	No	The departure station.

plaintext



| Destination | query | string | No | The arrival station. || BeginDate | query | string (date-time) | No | The starting date of the trip information search. || EndDate | query | string (date-time) | No | The end date of the trip information search. || StartTimeInterval | query | string (duration) | No | The starting time of the trip information search. || EndTimeInterval | query | string (duration) | No | The end time of the trip information search. || Identifier | query | string | No | The transportation identifier, such as a flight number. || CarrierCode | query | string | No | The carrier code. |

Responses

200 Response

404 Response

GET /api/nsk/v2/trip/info/simple

Summary: Performs a simplified search for trip information based upon minimal, required data for the search criteria.

Description

This endpoint will return a maximum of 100 journeys. If more than 100 journeys are needed, the search criteria will need to be modified and additional trip calls will need to be made.

Request Details

Parameters

Name	Location	Type	Required	Description
NumberOfJourneys	query	integer (int32)	No	When set to null, 100 journeys will be returned.
This endpoint will never return more than 100 journeys.				

plaintext



| Origin | query | string | No | The departure station. || Destination | query | string | No | The arrival station. || EndDate | query | string (date-time) | No | The end date of the trip information search. || StartTimeInterval | query | string (duration) | No | The starting time of the trip information search. || EndTimeInterval | query | string (duration) | No | The end time of the trip information search. || CarrierCode | query | string | No | The carrier code. || BeginDate | query | string (date-time) | No | The starting date of the trip information search. || Identifier | query | string | No | The transportation identifier, such as a flight number. || FlightType | query | string | No | The flight type.

Enumeration values: 0 = All, 1 = NonStop, 2 = Through, 3 = Direct, 4 = Connect |

Responses

200 Response

The trip information response.

404 Response

POST /api/nsk/v1/trip/info/status/operationDetails/status/mishapRestore

Summary: Restores locked passenger data associated the leg information. This restores the leg status to normal.

Description

Request Details

Request Body

The leg key.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

401 Response

Examples

Response 400

Unauthorized Agent:

```
json   
  
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Trip:MishapRestorePassengerDataUnauthorized",  
      "message": "Agent unauthorized to restore passenger data.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example of Agent unauthorized to restore passenger data for restore after mishap.

Invalid Carrier Code:

```
json   
  
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Trip:InvalidCarrierCode",  
      "message": "The carrier code 'x!*' is invalid.",  
      "type": "Error",  
      "details": null,  
      "rawMessage": null,  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example of invalid or unknown carrier code.

Invalid Origin:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Trip:InvalidOrigin",
      "message": "The Departure Station 'Gdo' is invalid.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example of invalid or unknown departure station.

Invalid Destination:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Trip:InvalidDestination",
      "message": "The Arrival Station 'hJA' is invalid.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example of invalid or unknown arrival station.

Invalid Flight Number:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Trip:InvalidFlightNumber",
      "message": "The flight number 'KYB9' is invalid.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example of an invalid flight number.

Mishap Restore Failed:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Trip:MishapRestoreFailed",
      "message": "NV1234 was not found or not restored..",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example of when restoring a flight from mishap failed.

POST /api/nsk/v2/trip/move

Summary: Moves a journey on the booking in state.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Request Body

Move request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v3/trip/move/availability

Summary: Searches for move availability with the full availability search request. This request will allow for total control over the configuration. If a property is not set, the default settings (AvailabilitySettings) is used in its place.

Description

Note, this is a post request even though it is only getting information. This is due to the complicated data needed in the post data.

Request Details

Request Body

The availability move request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

The availability response.

404 Response

GET /api/nsk/v3/trip/move/availability/{journeyKey}

Summary: Searches for availability with the simple move request. The simple search request covers the bare minimum search criteria and lets default settings (AvailabilitySettings) configure the rest.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
journeyKey	path	string	Yes	The journey key.
BeginDate	query	string (date-time)	No	The departure date.

```
plaintext 
```

|

| Origin | query | string | No | The origin station code. || Destination | query | string | No | The destination station code. || PassengerMoveType | query | string | No | The passenger move type.

Enumeration values: 0 = None, 1 = Irop, 2 = Diversion, 4 = FlightClose, 5 = FlyAhead, 6 = SplitJourney, 7 = SelfServiceRebooking |

Responses

200 Response

The availability response.

404 Response

GET /api/nsk/v2/trip/move/availability/selfService

Summary: Self-service search for move availability.

Description

If no BeginDate is provided, it will default to the journey departure date. If no EndDate is provided, it will default to the begin date + 1 day.

Request Details

Parameters

Name	Location	Type	Required	Description
BeginDate	query	string (date-time)	No	The begin date.

```
plaintext 
```

|

| EndDate | query | string (date-time) | No | The end date. |

Responses

200 Response

The availability response.

404 Response

POST /api/nsk/v1/trip/move/irop

Summary: Performs an IROP move for the booking in state. This endpoint adds additional IROP move functionality for agents. See [/api/nsk/v2/trip/move](#) for IROP functionality not restricted to agents only.

Description

Request Details

Request Body

IROP move request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v1/trip/passengers

Summary: Creates a new booking with the provided passengers.

Description

This endpoint is not for updating passengers on an existing booking. For that purpose, use the "booking/passenger/{passengerKey}" endpoints for PUT and PATCH.

plaintext



IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

Request Details

Parameters

Name	Location	Type	Required	Description
syncPassengerAndTravelDocGender	query	boolean	No	Indicates whether to synchronize the passenger and all its travel documents' gender. This will update the gender of all the travel documents in the passengers[passengerKey].travelDocuments list in the booking model. The value defaults to true if not specified.

plaintext



|

Request Body

The passengers to be added.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

404 Response

The item was not found.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 400

Missing passenger information:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:PassengerInfoIncomplete",
      "message": "PassengerInfo is required.",
      "type": "Error",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error for missing passenger information.

POST /api/nsk/v1/trip/passiveSegments

Summary: Creates a passive segment and creates a booking if it doesn't already exist.

Description

This endpoint creates a new journey with the passive segment on it. A passive segment cannot be updated, however get and delete work like all other journeys.

```
plaintext

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Request Body

The passive segment request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

POST /api/nsk/v4/trip/rebook/availability

Summary: Rebook search with the current booking defaults.

Description

Requires a booking in state. Although this endpoint behaves like a GET, POST is used to reduce the limitations resulting from URL length issues. The AvailabilityCriteriaFilter.FareInclusionType.NoPricing enumeration value, if set, will result in an availability response that does not include fares. The primary use case for this is in Move flows, but is being maintained on the main availability and rebook availability flows in case there are other use cases as well.

Request Details

Request Body

The rebook request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

GET /api/nsk/v4/trip/rebook/availability/simple

Summary: Simple rebook search with the current booking defaults.

Description

Requires a booking in state. The AvailabilityCriteriaFilter.FareInclusionType.NoPricing enumeration value, if set, will result in an availability response that does not include fares. The primary use case for this is in Move flows, but is being maintained on the main availability and rebook availability flows in case there are other use cases as well.

Request Details

Parameters

Name	Location	Type	Required	Description
Origin	query	string	No	The origin station code.

```
plaintext 
```

| Destination | query | string | No | The destination station code. || BeginDate | query | string (date-time) | No | The departure date. || EndDate | query | string (date-time) | No | The return date if this is a round trip request. || LoyaltyFilter | query | string | No | The loyalty fare filter.

Enumeration values: 0 = MonetaryOnly, 1 = PointsOnly, 2 = PointsAndMonetary, 3 = PreserveCurrent |

Responses

200 Response

404 Response

GET /api/nsk/v1/trip/schedule

Summary: Retrieves the schedule of flights for a market.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
Origin	query	string	No	The origin station code.

```
plaintext 
```

| Destination | query | string | No | The destination station code. || BeginDate | query | string (date-time) | No | The begin date to search for. || EndDate | query | string (date-time) | No | The end date to search for. || Type | query | string | No | The flight type filter.

Enumeration values: 0 = None, 1 = NonStop, 2 = Through, 3 = Direct, 4 = Connect, 5 = All |

Responses

200 Response

POST /api/nsk/v1/trip/schedule/adHoc

Summary: Creates an ad hoc flight.

Description

Request Details

Request Body

The commit itinerary variation request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

POST /api/nsk/v4/trip/sell

Summary: Sells the list of journeys and creates a booking if it doesn't already exist.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

This endpoint will ignore the contents of 'TripSellRequestv2.Passengers.Types.Details' if there are already passengers in the booking in state and use the existing passengers instead.

Request Details

Request Body

The trip sell request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

404 Response

The item was not found.

Examples

Response 200

Trip sold successfully.:

json



```
{
  "data": {
    "selfServiceMoveAvailable": false,
    "bookingKey": "MyE2dwP4ZSFmYWxzZQ--",
    "recordLocator": null,
    "currencyCode": null,
    "systemCode": null,
    "groupName": null,
    "locators": null,
    "info": null,
    "sales": null,
    "typeOfSale": null,
    "hold": null,
    "breakdown": null,
    "receivedBy": null,
    "contacts": null,
    "passengers": {
      "mcfbrfQ-": {
        "passengerKey": "MCFBRFQ-",
        "passengerAlternateKey": null,
        "customerNumber": null,
        "fees": null,
        "name": null,
        "passengerTypeCode": "ADT",
        "discountCode": null,
        "bags": null,
        "program": null,
        "infant": null,
        "info": null,
        "travelDocuments": null,
        "addresses": [
          {
            "status": 0,
            "companyName": null,
            "lineOne": "123 Fake Street",
            "passengerAddressKey": null,
            "phone": "555-555-1234",
            "lineTwo": null,
            "stationCode": "SLC",
            "lineThree": null,
            "emailAddress": null,
            "countryCode": "US",
            "cultureCode": null,
            "provinceState": "UT",
            "refusedContact": null,
            "city": "SLC",
            "postalCode": "84108"
          }
        ],
        "weightCategory": null,
        "emdCoupons": null
      }
    },
    "journeys": [
      {
        "flightType": 1,
        "stops": 0,
        "designator": {
          "destination": "DEN",
          "origin": "SLC",
          "arrival": "2025-08-08T00:00:00+00:00",
          "departure": "2025-08-07T00:00:00+00:00"
        },
        "move": null,
        "segments": [
          {
            "isStandby": false,
            "isConfirming": false,
            "isConfirmingExternal": false,
            "isBlocked": false,
            "isHosted": true,
            "isArrival": false
          }
        ]
      }
    ]
  }
}
```

```
"isChangeOfGauge": false,
"designator": {
    "destination": "DEN",
    "origin": "SLC",
    "arrival": "2025-08-08T00:00:00+00:00",
    "departure": "2025-08-07T00:00:00+00:00"
},
"isSeatmapViewable": true,
"fares": [
    {
        "isGoverning": true,
        "carrierCode": null,
        "fareKey": "MH5Cfn50Vn5CRU1EfkVTQ1J_fjB_MH5_WA--",
        "classOfService": null,
        "classType": null,
        "fareApplicationType": 0,
        "fareClassOfService": null,
        "fareBasisCode": null,
        "fareSequence": 0,
        "inboundOutBound": 0,
        "fareStatus": 0,
        "isAllotmentMarketFare": false,
        "originalClassOfService": null,
        "ruleNumber": null,
        "productClass": null,
        "ruleTariff": null,
        "travelClassCode": null,
        "pricingDate": "0001-01-01T00:00:00",
        "crossReferenceClassOfService": null,
        "fareDesignator": null,
        "passengerFares": null,
        "fareLink": 0
    }
],
"segmentKey": "MTIzNDEyMzQ-",
"identifier": {
    "identifier": "1234",
    "carrierCode": "NV",
    "opSuffix": null
},
"passengerSegment": {
    "mcfbrfq-": {
        "seats": null,
        "passengerKey": null,
        "activityDate": null,
        "boardingSequence": null,
        "createdDate": null,
        "liftStatus": 2,
        "modifiedDate": null,
        "overBookIndicator": 0,
        "priorityDate": null,
        "timeChanged": false,
        "verifiedTravelDocs": null,
        "sourcePointOfSale": null,
        "pointOfSale": null,
        "ssrs": [
            {
                "isConfirmed": false,
                "isConfirmingUnheld": false,
                "note": null,
                "ssrDuration": 0,
                "ssrKey": null,
                "count": 0,
                "ssrCode": "Ssr1",
                "feeCode": null,
                "inBundle": false,
                "passengerKey": null,
                "ssrDetail": null,
                "ssrNumber": null,
                "market": null
            }
        ],
        "ssrCount": 1
    }
}
```

```

        "tickets": null,
        "bags": null,
        "scores": null,
        "boardingPassDetail": null,
        "hasInfant": false,
        "seatPreferences": null,
        "bundleCode": null,
        "verifiedTravelDocuments": null,
        "referenceNumber": 0,
        "baggageGroupNumber": null
    }
},
"channelType": 0,
"cabinOfService": null,
"externalIdentifier": null,
"priorityCode": null,
"changeReasonCode": 0,
"segmentType": 0,
"salesDate": null,
"international": false,
"flightReference": null,
"legs": [
{
    "legKey": null,
    "operationsInfo": null,
    "designator": {
        "destination": "DEN",
        "origin": "SLC",
        "arrival": "2025-08-08T00:00:00+00:00",
        "departure": "2025-08-07T00:00:00+00:00"
    },
    "legInfo": null,
    "nests": null,
    "ssrs": null,
    "seatmapReference": null,
    "flightReference": null
}
],
"status": 0
}
],
"journeyKey": "MjM0MjRhZmFzZGY-",
"notForGeneralUser": false
}
],
"comments": null,
"queues": null,
"history": null,
"payments": null,
"addOns": null,
"orders": null
}
}

```

Description: Example response when the trip is sold successfully.

PaxPriceDetails are ignored.:

json



```
{
  "messages": [
    {
      "code": "nsk:Trip:PaxPriceDetailsRequestIgnored",
      "type": "Warning",
      "value": "The contents of 'TripSellRequestv2.Passengers.Types.Details' are ignored because the booking in state already contains a value for this field. This is likely a bug in the application logic. Please review the code and fix it if necessary.",
      "status": 0,
      "details": null
    }
  ],
  "data": {
    "selfServiceMoveAvailable": false,
    "bookingKey": "MyE2dWp4ZSFmYWxzZQ--",
    "recordLocator": null,
    "currencyCode": null,
    "systemCode": null,
    "groupName": null,
    "locators": null,
    "info": null,
    "sales": null,
    "typeOfSale": null,
    "hold": null,
    "breakdown": null,
    "receivedBy": null,
    "contacts": null,
    "passengers": {
      "mcfbrfQ-": {
        "passengerKey": "MCFBRFQ-",
        "passengerAlternateKey": null,
        "customerNumber": null,
        "fees": null,
        "name": null,
        "passengerTypeCode": "ADT",
        "discountCode": null,
        "bags": null,
        "program": null,
        "infant": null,
        "info": null,
        "travelDocuments": null,
        "addresses": [
          {
            "status": 0,
            "companyName": null,
            "lineOne": "123 Fake Street",
            "passengerAddressKey": null,
            "phone": "555-555-1234",
            "lineTwo": null,
            "stationCode": "SLC",
            "lineThree": null,
            "emailAddress": null,
            "countryCode": "US",
            "cultureCode": null,
            "provinceState": "UT",
            "refusedContact": null,
            "city": "SLC",
            "postalCode": "84108"
          }
        ],
        "weightCategory": null,
        "emdCoupons": null
      }
    },
    "journeys": [
      {
        "flightType": 1,
        "stops": 0,
        "designator": {
          "destination": "DEN",
          "origin": "SLC",
          "arrival": "2025-08-08T00:00:00+00:00",
          "departure": "2025-08-07T00:00:00+00:00"
        }
      }
    ]
  }
}
```

```
        },
        "move": null,
        "segments": [
            {
                "isStandby": false,
                "isConfirming": false,
                "isConfirmingExternal": false,
                "isBlocked": false,
                "isHosted": true,
                "isChangeOfGauge": false,
                "designator": {
                    "destination": "DEN",
                    "origin": "SLC",
                    "arrival": "2025-08-08T00:00:00+00:00",
                    "departure": "2025-08-07T00:00:00+00:00"
                },
                "isSeatmapViewable": true,
                "fares": [
                    {
                        "isGoverning": true,
                        "carrierCode": null,
                        "fareKey": "MH5Cfn50Vn5CRU1EfkVTQ1J_fjB_MH5_WA--",
                        "classOfService": null,
                        "classType": null,
                        "fareApplicationType": 0,
                        "fareClassOfService": null,
                        "fareBasisCode": null,
                        "fareSequence": 0,
                        "inboundOutBound": 0,
                        "fareStatus": 0,
                        "isAllotmentMarketFare": false,
                        "originalClassOfService": null,
                        "ruleNumber": null,
                        "productClass": null,
                        "ruleTariff": null,
                        "travelClassCode": null,
                        "pricingDate": "0001-01-01T00:00:00",
                        "crossReferenceClassOfService": null,
                        "fareDesignator": null,
                        "passengerFares": null,
                        "fareLink": 0
                    }
                ],
                "segmentKey": "MTIzNDEyMzQ-",
                "identifier": {
                    "identifier": "1234",
                    "carrierCode": "NV",
                    "opSuffix": null
                },
                "passengerSegment": {
                    "mcfbrfQ-": {
                        "seats": null,
                        "passengerKey": null,
                        "activityDate": null,
                        "boardingSequence": null,
                        "createdDate": null,
                        "liftStatus": 2,
                        "modifiedDate": null,
                        "overBookIndicator": 0,
                        "priorityDate": null,
                        "timeChanged": false,
                        "verifiedTravelDocs": null,
                        "sourcePointOfSale": null,
                        "pointOfSale": null,
                        "ssrs": [
                            {
                                "isConfirmed": false,
                                "isConfirmingUnheld": false,
                                "note": null,
                                "ssrDuration": 0,
                                "ssrKey": null,
                                "count": 0,
                                "isUnheld": false
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```

```

        "ssrCode": "Ssr1",
        "feeCode": null,
        "inBundle": false,
        "passengerKey": null,
        "ssrDetail": null,
        "ssrNumber": null,
        "market": null
    }
],
"tickets": null,
"bags": null,
"scores": null,
"boardingPassDetail": null,
"hasInfant": false,
"seatPreferences": null,
"bundleCode": null,
"verifiedTravelDocuments": null,
"referenceNumber": 0,
"baggageGroupNumber": null
}
},
"channelType": 0,
"cabinOfService": null,
"externalIdentifier": null,
"priorityCode": null,
"changeReasonCode": 0,
"segmentType": 0,
"salesDate": null,
"international": false,
"flightReference": null,
"legs": [
{
"legKey": null,
"operationsInfo": null,
"designator": {
"destination": "DEN",
"origin": "SLC",
"arrival": "2025-08-08T00:00:00+00:00",
"departure": "2025-08-07T00:00:00+00:00"
},
"legInfo": null,
"nests": null,
"ssrs": null,
"seatmapReference": null,
"flightReference": null
}
],
"status": 0
}
],
"journeyKey": "MjM0MjRhZmFzZGY-",
"notForGeneralUser": false
}
],
"comments": null,
"queues": null,
"history": null,
"payments": null,
"addOns": null,
"orders": null
}
}
}

```

Description: Example message when there are already existing booking passengers and the Passengers.Types. The trip is sold successfully and the details in the request are ignored.

POST /api/nsk/v1/trip/standby/availability

Summary: Retrieves same-day standby availability relevant to the current state booking.

Description

This endpoint only provisions same-day standby availability. The existing, standard flight availability endpoints should still be used for other standby availability search scenarios. Although this endpoint behaves like a GET, POST reduces the limitations resulting from URL length issues.

Request Details

Request Body

The standby availability request to use in the search.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

404 Response

User Endpoints

GET /api/nsk/v1/user

Summary: Retrieves the current logged in user.

Description

Request Details

Responses

404 Response

200 Response

The user.

POST /api/nsk/v1/user

Summary: Creates a user and person (used by anonymous to create non agent logins).

Description

This endpoint will create a user based off the system configured customer creation parameters. The associated person has to be of type customer or an exception will be thrown. To change these default settings please configure utilities.

Request Details

Request Body

The user customer create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

PUT /api/nsk/v1/user

Summary: Updates the current logged in user.

Description

Request Details

Request Body

The user edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user

Summary: Patches the current logged in user.

Description

Request Details

Request Body

The user patch request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/user/bookings

Summary: Searches the logged in user booking upcoming and past bookings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
AssociatedFirstName	query	string	No	An associated passenger/contact first name.

plaintext ✖

|

| AssociatedLastName | query | string | No | An associated passenger/contact last name. || PhoneticSearch | query | boolean | No | Flag indicating to search using the phonetic option. || TripIdentifier | query | string | No | Filter by trip identifier. || Origin | query | string | No | Filter by origin station code. || Destination | query | string | No | Filter by destination station code. || SearchArchive | query | boolean | No | Flag indicating to search archived bookings. || TripDate | query | string (date-time) | No | Filter by trip date. || StartDate | query | string (date-time) | No | Filter by starting search date. || EndDate | query | string (date-time) | No | Filter by ending search date. || SearchByCustomerNumber | query | boolean | No | Flag indicating to search by the users customer number instead of user identifier. || ReturnCount | query | integer | No | The number of results to be returned. || LastBookingKey | query | string | No | The last booking key. || StartingIndex | query | integer (int64) | No | The starting index of the next set of bookings (used for paging). |

Responses

404 Response

200 Response

The collection of bookings.

GET /api/nsk/v1/user/bookingsByContactCustomerNumber

Summary: Searches for any bookings that have the current user as a contact via their customer number.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
PageSize	query	integer	No	The number of items to return for the request.

```
plaintext 
```

| LastIndex | query | integer (int64) | No | The last booking index (used for paging). || FlightNumber | query | string | No | Filter by the flight number. |
| DepartureDate | query | string (date-time) | No | The first journeys departure date. || Destination | query | string | No | Filter by the destination code. || Origin | query | string | No | Filter by the departure code. || SourceOrganization | query | string | No | Filter by the organization associated with the booking. For example the travel agency associated with this booking. || OrganizationGroupCode | query | string | No | Filter by the organization group code to find. || SearchArchive | query | boolean | No | Whether or not to search the booking archive. |

Responses

200 Response

The collection of bookings.

GET /api/nsk/v1/user/bookingsByPassenger

Summary: Searches the logged in user booking upcoming and past bookings.

Description

This is similar to the user booking endpoint but returns more focused trip information. This endpoint will only return data where a booking's passenger customer number matches the current logged in user. Bookings made by the user where the passenger customer number is not assigned are ignored. The BeginDate will default to today, and the EndDate defaults to one month from today unless a value is provided.

Request Details

Parameters

Name	Location	Type	Required	Description
StartDate	query	string (date-time)	No	Booking start search date.

```
plaintext 
```

| EndDate | query | string (date-time) | No | Booking end search date. |

Responses

200 Response

The collection of bookings.

GET /api/nsk/v1/user/impersonate

Summary: Gets the logged in users current session roles state.

Description

Request Details

Responses

404 Response

200 Response

The current session roles.

POST /api/nsk/v1/user/impersonate

Summary: Impersonates a new role for the logged in user.

Description

Request Details

Request Body

The user impersonate request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

202 Response

Ok.

DELETE /api/nsk/v1/user/impersonate

Summary: Resets the logged in users role.

Description

Request Details

Responses

400 Response

200 Response

Ok.

Examples

Response 200

Success.:

```
json
{
  "data": null
}
```



Description: Success will return a 200 with no data.

Not impersonating.:

```
json

{
  "messages": [
    {
      "code": "nsk:NoOperation",
      "type": "Warning",
      "value": "No operation was performed. The data is already in the expected state.",
      "status": 0,
      "details": null
    }
  ],
  "data": null
}
```

Description: If not needed a 200 will be returned with a warning.

POST /api/nsk/v3/user/password/change

Summary: Changes the logged in user's password.

Description

Validates that the current password is correct before proceeding.

```
plaintext

This follows domain level restrictions and could result in a failed change if there is a minimum time
requirement between password changes, invalid password length, or invalid character.
```

Request Details

Request Body

The change password request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Request

UserChangePasswordRequestv2:

```
json

{
  "currentPassword": "oldPassword123!",
  "newPassword": "newPassword123!"
}
```

Description: Updates the current user's password with the new provided one.

Response 400

Error response for incorrect domain, username, or password.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:AgentAuthentication: The agent (WWW/ABC123) was not authenticated",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode.

Error response for a locked account:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:AgentLocked: The agent (WWW/ABC123) is locked.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode.

Error response for minimum password time:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:PasswordMinimumTime: The agent (WWW/ABC123) cannot change their password before the Minimum Time is reached",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode. Minimum time is determined by domain configuration.

Error response for password policy violation.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:PasswordChange:Failed",
      "message": null,
      "type": "Error",
      "details": null,
      "rawMessage": "nsk-server:PasswordPolicy: X, Y, Z",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode. Password policy is configurable and determined by the domain.

Error response when the new password is already used as password before.:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:PasswordHistory",
      "message": "nsk-server:PasswordHistory",
      "type": "Information",
      "details": null,
      "rawMessage": "The agent (WWW/tst01) cannot reuse this password.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Only returns raw message in debug mode. Unique password count is determined by domain configuration.

GET /api/nsk/v1/user/roles

Summary: Gets all roles from the logged in user.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of roles.

POST /api/nsk/v1/user/roles

Summary: Creates a new role on the logged in user.

Description

Request Details

Request Body

The new role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/roles/{userRoleKey}

Summary: Gets a specific role from the logged in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userRoleKey	path	string	Yes	The unique user role key

Responses

404 Response

200 Response

The user role.

PUT /api/nsk/v1/user/roles/{userRoleKey}

Summary: Updates a specific role on the logged in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userRoleKey	path	string	Yes	The unique user role key.

Request Body

The modified role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/roles/{userRoleKey}

Summary: Patches a specific role on the logged in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userRoleKey	path	string	Yes	The unique user role key.

Request Body

The patched role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/roles/{userRoleKey}

Summary: Deletes a specific role on the logged in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userRoleKey	path	string	Yes	The unique user role key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/singleSignOnToken

Summary: Gets all of the single sign on tokens associated with a specific user.

Description

Request Details

Responses

404 Response

200 Response

A collection of single sign on tokens associated with the user.

GET /api/nsk/v1/user/singleSignOnToken/{providerKey}

Summary: Gets a single sign on token associated with a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerKey	path	string	Yes	The single sign on provider key.

Responses

404 Response

200 Response

A single sign on token.

POST /api/nsk/v1/user/singleSignOnToken/{providerKey}

Summary: Links the requested token with the logged-in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerKey	path	string	Yes	The single sign on provider key.

Request Body

The single sign on token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

PUT /api/nsk/v1/user/singleSignOnToken/{providerKey}

Summary: Updates an existing token associated with the logged-in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerKey	path	string	Yes	The single sign on provider key.

Request Body

The single sign on token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/user/singleSignOnToken/{providerKey}

Summary: Patches an existing token associated with the logged-in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerKey	path	string	Yes	The single sign on provider key.

Request Body

The single sign on token request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/user/singleSignOnToken/{providerKey}

Summary: Deletes a token associated with the single sign on provider from the logged-in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
providerKey	path	string	Yes	The single sign on provider key.

Responses

200 Response

User/Person Endpoints

GET /api/nsk/v1/user/person

Summary: Retrieves the current logged in person.

Description

Request Details

Responses

404 Response

200 Response

The person.

PUT /api/nsk/v1/user/person

Summary: Updates the current logged in person record basic information.

Description

Request Details

Request Body

The person edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person

Summary: Patches the current logged in person records basic information.

Description

Request Details

Request Body

The person patch request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/user/person/account

Summary: Retrieves the currently logged in person's account and collections.

Description

Request Details

Responses

404 Response

200 Response

GET /api/nsk/v2/user/person/account/collection/{accountCollectionKey}/transactions

Summary: Retrieves the currently logged in person's account collection transactions..

Description

Request Details

Parameters

Name	Location	Type	Required	Description
accountCollectionKey	path	string	Yes	The account collection key.
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

plaintext



| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

json



```
{  
  "data": {  
    "totalCount": 11085,  
    "lastPageKey": "MTAw",  
    "transactions": [  
      {  
        "accountTransactionCode": null,  
        "recordLocator": null,  
        "accountReference": null,  
        "transactionKey": "MSEXITEwMjA4OCJsJNQxATUzMyEhUURRNUZ",  
        "amount": -2.42,  
        "createdDate": "2025-08-07T04:47:18.3114853Z",  
        "currencyCode": "USD",  
        "foreignAmount": -15.0,  
        "note": "Funds Used: PNR123",  
        "foreignCurrencyCode": "CNY",  
        "type": 0,  
        "paymentId": 365423,  
        "accountCollectionKey": "MSex"  
      }  
    ]  
  }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json
{
  "errors": [
    {
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "TransactionSearchRequestv2",
        "member": "StartDate"
      },
      "rawMessage": "The TransactionSearchRequestv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```
json
{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when end date is before start date.

GET /api/nsk/v2/user/person/account/transactions

Summary: Retrieves all transactions for the currently logged in person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
StartDate	query	string (date-time)	No	The starting date and time of the date range to search in.

```
plaintext
|
|
```

| EndDate | query | string (date-time) | No | The end date and time of the date range to search in. || SortByNewest | query | boolean | No | Optional filter to sort results by newest. || PageSize | query | integer | No | Used to specify the number of records to return. || LastPageKey | query | string | No | The last page key where to start paged query. |

Responses

200 Response

400 Response

Examples

Response 200

Retrieve Transactions Successful:

```
json 
```

```
{  
  "data": {  
    "totalCount": 11085,  
    "lastPageKey": "MTAw",  
    "transactions": [  
      {  
        "accountTransactionCode": null,  
        "recordLocator": null,  
        "accountReference": null,  
        "transactionKey": "MSExITEwMjA4OCJsJNQxATUzMyEhUURRNuz",  
        "amount": -2.42,  
        "createdDate": "2025-08-07T04:47:18.3098356Z",  
        "currencyCode": "USD",  
        "foreignAmount": -15.0,  
        "note": "Funds Used: PNR123",  
        "foreignCurrencyCode": "CNY",  
        "type": 0,  
        "paymentId": 365423,  
        "accountCollectionKey": "MSex"  
      }  
    ]  
  }  
}
```

Description: Example response for a successful retrieval of transactions.

Response 400

Empty Start Date:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": "154b85c0-1119-42c2-895c-1aa09ec3a85b",  
      "code": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "message": "TransactionSearchRequestv2.StartDate:RequiredAttribute",  
      "type": "Validation",  
      "details": {  
        "validation": "RequiredAttribute",  
        "model": "TransactionSearchRequestv2",  
        "member": "StartDate"  
      },  
      "rawMessage": "The TransactionSearchRequestv2 field is required.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when start date field is empty.

End Date Before Start Date:

```
json

{
  "errors": [
    {
      "id": "4e33c648-abec-fa59-0877-5e0bfee7ea87",
      "code": "nsk:Validation:EndDateBeforeStartDate",
      "message": "nsk:Validation:EndDateBeforeStartDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 1/1/2011 occurs before the requested start date of 1/1/2012.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example response when end date is before start date.

GET /api/nsk/v1/user/person/addresses

Summary: Gets all addresses from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

POST /api/nsk/v1/user/person/addresses

Summary: Creates a new address on the logged in users person.

Description

Request Details

Request Body

The new address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/addresses/{personAddressKey}

Summary: Gets a specific address from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAddressKey	path	string	Yes	The unique person address key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/user/person/addresses/{personAddressKey}

Summary: Updates a specific address on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The modified address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/user/person/addresses/{personAddressKey}

Summary: Patches a specific address on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The patched address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/user/person/addresses/{personAddressKey}

Summary: Deletes a specific address on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAddressKey	path	string	Yes	The unique person address key.

Responses

200 Response

GET /api/nsk/v1/user/person/affiliations

Summary: Gets all affiliations from the logged in users person.

Description

Request Details

Responses

404 Response

200 Response

POST /api/nsk/v1/user/person/affiliations

Summary: Creates a new affiliation on the logged in users person.

Description

Request Details

Request Body

The new affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/affiliations/{personAffiliationKey}

Summary: Gets a specific affiliation from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/user/person/affiliations/{personAffiliationKey}

Summary: Updates a specific affiliation on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Request Body

The modified affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/user/person/affiliations/{personAffiliationKey}

Summary: Deletes a specific affiliation on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

200 Response

GET /api/nsk/v1/user/person/aliases

Summary: Gets all aliases from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of aliases.

POST /api/nsk/v1/user/person/aliases

Summary: Creates a new alias on the logged in users person.

Description

Request Details

Request Body

The new alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/aliases/{personAliasKey}

Summary: Gets a specific alias from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAliasKey	path	string	Yes	The unique person alias key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/user/person/aliases/{personAliasKey}

Summary: Updates a specific alias on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The modified alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/aliases/{personAliasKey}

Summary: Patches a specific alias on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The patched alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/aliases/{personAliasKey}

Summary: Deletes a specific alias on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personAliasKey	path	string	Yes	The unique person alias key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/comments

Summary: Gets all comments from the logged in users person.

Description

Request Details

Responses

404 Response

200 Response

The collection of comments.

POST /api/nsk/v1/user/person/comments

Summary: Creates a new comment on the logged in users person.

Description

Request Details

Request Body

The new comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/comments/{personCommentKey}

Summary: Gets a specific comment from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personCommentKey	path	string	Yes	The unique person comment key.

Responses

404 Response

200 Response

The comment.

PUT /api/nsk/v1/user/person/comments/{personCommentKey}

Summary: Updates a specific comment on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The modified comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/comments/{personCommentKey}

Summary: Patches a specific comment on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The patched comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/comments/{personCommentKey}

Summary: Deletes a specific comment on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personCommentKey	path	string	Yes	The unique person comment key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/emails

Summary: Gets all email addresses from the logged in user's person.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of email addresses.

POST /api/nsk/v1/user/person/emails

Summary: Creates a new email address on the logged in user's person.

Description

The "type" property for the request is a single character. See the "/resources/EmailTypes" endpoint for valid email type codes.

Request Details

Request Body

The new email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/emails/{personEmailAddressKey}

Summary: Gets a specific email address from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personEmailAddressKey	path	string	Yes	The unique person email address key.

Responses

404 Response

200 Response

The email address.

PUT /api/nsk/v1/user/person/emails/{personEmailAddressKey}

Summary: Updates a specific email address on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The modified email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/emails/{personEmailAddressKey}

Summary: Patches a specific email address on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The patched email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/emails/{personEmailAddressKey}

Summary: Deletes a specific email address on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personEmailAddressKey	path	string	Yes	The unique person email address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/information

Summary: Gets all information from the logged in user's person.

Description

Request Details

Responses

404 Response

200 Response

The collection of information.

POST /api/nsk/v1/user/person/information

Summary: Creates a new information on the logged in user's person.

Description

Request Details

Request Body

The new information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/information/{personInformationKey}

Summary: Gets a specific information from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personInformationKey	path	string	Yes	The unique person information key.

Responses

404 Response

200 Response

The information.

PUT /api/nsk/v1/user/person/information/{personInformationKey}

Summary: Updates a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The modified information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/information/{personInformationKey}

Summary: Patches a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The patched information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/information/{personInformationKey}

Summary: Deletes a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personInformationKey	path	string	Yes	The unique person information key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/phoneNumbers

Summary: Gets all phone numbers from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

POST /api/nsk/v1/user/person/phoneNumbers

Summary: Creates a new phone number on the logged in users person.

Description

Request Details

Request Body

The new phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey}

Summary: Gets a specific phone number from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey}

Summary: Updates a specific phone number on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The modified phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey}

Summary: Patches a specific phone number on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The patched phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/user/person/phoneNumbers/{personPhoneNumberKey}

Summary: Deletes a specific phone number on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Responses

200 Response

GET /api/nsk/v1/user/person/preferences

Summary: Gets all preferences from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of preferences.

POST /api/nsk/v1/user/person/preferences

Summary: Creates a new preference on the logged in users person.

Description

Request Details

Request Body

The new preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/preferences/{personPreferenceKey}

Summary: Gets a specific preference from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

404 Response

200 Response

The preference.

PUT /api/nsk/v1/user/person/preferences/{personPreferenceKey}

Summary: Updates a specific preference on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The modified preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/preferences/{personPreferenceKey}

Summary: Patches a specific preference on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The patched preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/preferences/{personPreferenceKey}

Summary: Deletes a specific preference on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/programs

Summary: Gets all programs from the logged in users person.

Description

Request Details

Responses

404 Response

200 Response

The collection of programs.

POST /api/nsk/v1/user/person/programs

Summary: Creates a new program on the logged in users person.

Description

Request Details

Request Body

The new program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/programs/{personProgramKey}

Summary: Gets a specific program from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personProgramKey	path	string	Yes	The unique person program key.

Responses

404 Response

200 Response

The program.

PUT /api/nsk/v1/user/person/programs/{personProgramKey}

Summary: Updates a specific program on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The modified program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/programs/{personProgramKey}

Summary: Patches a specific program on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The patched program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/programs/{personProgramKey}

Summary: Deletes a specific program on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personProgramKey	path	string	Yes	The unique person address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/storedPayments

Summary: Gets all stored payments from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of stored payments.

POST /api/nsk/v1/user/person/storedPayments

Summary: Creates a new stored payment on the logged in users person.

Description

Request Details

Request Body

The new stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey}

Summary: Gets a specific stored payment from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Responses

404 Response

200 Response

The stored payment.

PUT /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey}

Summary: Updates a specific stored payment on the logged in users person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The modified stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey}

Summary: Patches a specific stored payment on the logged in users person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The patched stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/user/person/storedPayments/{personStoredPaymentKey}

Summary: Deletes a specific stored payment on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Responses

200 Response

Ok.

GET /api/nsk/v1/user/person/travelDocuments

Summary: Gets all travel documents from the logged in users person.

Description

No description available

Request Details

Responses

404 Response

200 Response

The collection of travel documents.

POST /api/nsk/v1/user/person/travelDocuments

Summary: Creates a new travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Request Body

The new travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/user/person/travelDocuments/{personTravelDocumentKey}

Summary: Gets a specific travel document from the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Responses

404 Response

200 Response

The travel document.

DELETE /api/nsk/v1/user/person/travelDocuments/{personTravelDocumentKey}

Summary: Deletes a specific travel document on the logged in users person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Responses

200 Response

Ok.

PUT /api/nsk/v2/user/person/travelDocuments/{personTravelDocumentKey}

Summary: Updates a specific travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The modified travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v2/user/person/travelDocuments/{personTravelDocumentKey}

Summary: Patches a specific travel document on the logged in users person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The patched travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/user/person/travelNotifications

Summary: Gets person travel notifications for the logged in user's person.

Description

Request Details

Responses

404 Response

200 Response

POST /api/nsk/v1/user/person/travelNotifications

Summary: Creates a person travel notification for the logged in user's person.

Description

Request Details

Request Body

The travel notification create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}

Summary: Gets a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}

Summary: Updates a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}

Summary: Patches a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The delta mapper travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}

Summary: Deletes a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

GET /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events

Summary: Gets notification events for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events

Summary: Creates a notification event for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification event create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Gets a notification event for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

404 Response

DELETE /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Deletes a notification event from a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The notification event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

GET /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Gets notification timed events for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST /api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Creates a notification timed event for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification timed even create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}

Summary: Gets a notification timed event for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

404 Response

PUT

/api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}

Summary: Updates a notification timed event for a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Request Body

The updated notification timed event request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE

/api/nsk/v1/user/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventType}

Summary: Deletes a notification timed event from a person travel notification for the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

Users Endpoints

POST /api/nsk/v1/users

Summary: Creates a new user and person (used by agents to create other users).

Description

This method is used by agents to create other users (either agents or customers). Use this method when you want to create both a user and a person at the same time.

Request Details

Request Body

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 400

With Duplicate Agent Name:

```
json
{
  "errors": [
    {
      "id": "5d7a5d1c-d481-9723-260d-59f53b861112",
      "code": "nsk:User:UserNameAlreadyExists",
      "message": "Agent name 'firstAgent' already exists.",
      "type": "Information",
      "details": null,
      "rawMessage": "Agent name 'firstAgent' already exists.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ]
}
```

Description: The agent name already exists.

With Non-Existent Organization:

```
json
{
  "errors": [
    {
      "id": "17df744c-6115-9771-9151-5e9faa488e44",
      "code": "nsk-server:OrganizationNotFound",
      "message": "The requested organization was not found. Organization ID = '0999''",
      "type": "Information",
      "details": null,
      "rawMessage": "The requested organization was not found. Organization ID = '0999''",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ]
}
```

Description: The organization is not found on the server.

GET /api/nsk/v2/users

Summary: Search user records.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
OrganizationCode	query	string	No	The organization code to search by.

plaintext 

| LocationGroupCode | query | string | No | The location group code to search by. || DomainCode | query | string | No | The domain code to search in. || Username | query | string | No | The unique username to search by. || UsernameMatching | query | string | No | The matching criteria for the username.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || FirstName | query | string | No | The agent's first name to search by. || FirstNameMatching | query | string | No | The matching criteria for the agent's first name.

Enumeration values: 0 = StartsWith, 1 = EndsWith, 2 = Contains, 3 = ExactMatch || LastName | query | string | No | The agent's last name to search by. || RoleCode | query | string | No | The role code to search with. || Status | query | string | No | The agent status to search by.

Enumeration values: 0 = Default, 1 = Active, 2 = Pending, 3 = Suspended, 4 = Terminated || ReturnCount | query | integer | No | The number of results to return. || LastIndex | query | integer (int64) | No | The last user index (used for paging). |

Responses

404 Response

The item was not found.

200 Response

OK. Request completed successfully.

POST /api/nsk/v2/users

Summary: Creates a new user for a person (used by agents to create other users).

Description

This method is used by agents to create other users (either agents or customers) and connect them with an existing person. Use this method when a person already exists and you want to create a user for them.

Request Details

Request Body

The request for the user to be created.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

The item was created successfully.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 400

With Duplicate Agent Name:

```
json
{
  "errors": [
    {
      "id": "5d7a5d1c-d481-9723-260d-59f53b861112",
      "code": "nsk:User:UserNameAlreadyExists",
      "message": "Agent name 'firstAgent' already exists.",
      "type": "Information",
      "details": null,
      "rawMessage": "Agent name 'firstAgent' already exists.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The agent name already exists.

With Non-Existent Organization:

```
json
{
  "errors": [
    {
      "id": "17df744c-6115-9771-9151-5e9faa488e44",
      "code": "nsk-server:OrganizationNotFound",
      "message": "The requested organization was not found. Organization ID = '0999'",
      "type": "Information",
      "details": null,
      "rawMessage": "The requested organization was not found. Organization ID = '0999'",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

Description: The organization is not found on the server.

GET /api/nsk/v1/users/{userKey}

Summary: Retrieves the a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key of the user to retrieve.

Responses

404 Response

The item was not found.

200 Response

OK. Request completed successfully.

PUT /api/nsk/v1/users/{userKey}

Summary: Updates the specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The user edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

500 Response

A server error occurred.

Examples

Response 400

With Duplicate Agent Name:

```
json Copy  
{  
  "errors": [  
    {  
      "id": "5d7a5d1c-d481-9723-260d-59f53b861112",  
      "code": "nsk:User:UserNameAlreadyExists",  
      "message": "Agent name 'firstAgent' already exists.",  
      "type": "Information",  
      "details": null,  
      "rawMessage": "Agent name 'firstAgent' already exists.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ]  
}
```

Description: The agent name already exists.

With Non-Existent Organization:

```
json Copy  
{  
  "errors": [  
    {  
      "id": "17df744c-6115-9771-9151-5e9faa488e44",  
      "code": "nsk-server:OrganizationNotFound",  
      "message": "The requested organization was not found. Organization ID = '0999'",  
      "type": "Information",  
      "details": null,  
      "rawMessage": "The requested organization was not found. Organization ID = '0999'",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ]  
}
```

Description: The organization is not found on the server.

PATCH /api/nsk/v1/users/{userKey}

Summary: Patches the current logged in user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The user patch request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

OK. Request completed successfully.

400 Response

There was a problem with the request and it could not be completed.

500 Response

A server error occurred.

Examples

Response 400

With Duplicate Agent Name:

```
json   
{  
  "errors": [  
    {  
      "id": "5d7a5d1c-d481-9723-260d-59f53b861112",  
      "code": "nsk:User:UserNameAlreadyExists",  
      "message": "Agent name 'firstAgent' already exists.",  
      "type": "Information",  
      "details": null,  
      "rawMessage": "Agent name 'firstAgent' already exists.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ]  
}
```

Description: The agent name already exists.

With Non-Existent Organization:

```

json
{
  "errors": [
    {
      "id": "17df744c-6115-9771-9151-5e9faa488e44",
      "code": "nsk-server:OrganizationNotFound",
      "message": "The requested organization was not found. Organization ID = '0999'",
      "type": "Information",
      "details": null,
      "rawMessage": "The requested organization was not found. Organization ID = '0999'",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}

```

Description: The organization is not found on the server.

DELETE /api/nsk/v1/users/{userKey}

Summary: Delete the user.

Description

This is a soft delete by setting the user record status to terminated.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	User key.

Responses

200 Response

OK. Request completed successfully.

GET /api/nsk/v1/users/{userKey}/bookings

Summary: Searches the specific user for upcoming and past bookings.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
AssociatedFirstName	query	string	No	An associated passenger/contact first name.

```

plaintext
| AssociatedLastName | query | string | No | An associated passenger/contact last name. || PhoneticSearch | query | boolean | No | Flag indicating to search using the phonetic option. || TripIdentifier | query | string | No | Filter by trip identifier. || Origin | query | string | No | Filter by origin station code. || Destination | query | string | No | Filter by destination station code. || SearchArchive | query | boolean | No | Flag indicating to search archived bookings. || TripDate | query | string (date-time) | No | Filter by trip date. || StartDate | query | string (date-time) | No | Filter by starting search date. || EndDate | query | string (date-time) | No | Filter by ending search date. || SearchByCustomerNumber | query | boolean | No | Flag indicating to search by the users customer number instead of user identifier. || ReturnCount | query | integer | No | The number of results to be

```

| AssociatedLastName | query | string | No | An associated passenger/contact last name. || PhoneticSearch | query | boolean | No | Flag indicating to search using the phonetic option. || TripIdentifier | query | string | No | Filter by trip identifier. || Origin | query | string | No | Filter by origin station code. || Destination | query | string | No | Filter by destination station code. || SearchArchive | query | boolean | No | Flag indicating to search archived bookings. || TripDate | query | string (date-time) | No | Filter by trip date. || StartDate | query | string (date-time) | No | Filter by starting search date. || EndDate | query | string (date-time) | No | Filter by ending search date. || SearchByCustomerNumber | query | boolean | No | Flag indicating to search by the users customer number instead of user identifier. || ReturnCount | query | integer | No | The number of results to be

returned. || LastBookingKey | query | string | No | The last booking key. || StartingIndex | query | integer (int64) | No | The starting index of the next set of bookings (used for paging). |

Responses

404 Response

200 Response

The collection of bookings.

GET /api/nsk/v1/users/{userKey}/bookingsByPassenger

Summary: Searches the specific user upcoming and past bookings by passenger.

Description

This is similar to the user booking endpoint but returns more focused trip information. This endpoint will only return data where a bookings passenger customer number matches the current logged in user. Bookings made by the user but where the passenger customer are not assigned are ignored. The BeginDate will default to today, and the EndDate defaults to one month from today unless a value is provided.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
StartDate	query	string (date-time)	No	Booking start search date.

plaintext ✖

|

| EndDate | query | string (date-time) | No | Booking end search date. |

Responses

200 Response

The collection of bookings.

POST /api/nsk/v1/users/{userKey}/password/change

Summary: Changes a specifics user's password.

Description

Requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The change password request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Ok.

POST /api/nsk/v1/users/{userKey}/password/reset

Summary: Invokes the forgot password reset for a specific user.

Description

Requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

202 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person

Summary: Retrieves the a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	No description

Responses

404 Response

200 Response

The user's person.

PUT /api/nsk/v1/users/{userKey}/person

Summary: Updates the specific user's person record basic information.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The person edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person

Summary: Patches the specific user's person records basic information.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The person patch request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/users/{userKey}/person/addresses

Summary: Gets all addresses from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

POST /api/nsk/v1/users/{userKey}/person/addresses

Summary: Creates a new address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey}

Summary: Gets a specific address from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAddressKey	path	string	Yes	The unique person address key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey}

Summary: Updates a specific address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The modified address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey}

Summary: Patches a specific address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAddressKey	path	string	Yes	The unique person address key.

Request Body

The patched address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/addresses/{personAddressKey}

Summary: Deletes a specific address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAddressKey	path	string	Yes	The unique person address key.

Responses

200 Response

GET /api/nsk/v1/users/{userKey}/person/affiliations

Summary: Gets all affiliations from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

POST /api/nsk/v1/users/{userKey}/person/affiliations

Summary: Creates a new affiliation for the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey}

Summary: Gets a specific affiliation from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey}

Summary: Updates a specific affiliation for the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Request Body

The modified affiliation.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/affiliations/{personAffiliationKey}

Summary: Deletes a specific affiliation for the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAffiliationKey	path	string	Yes	The unique person affiliation key.

Responses

200 Response

GET /api/nsk/v1/users/{userKey}/person/aliases

Summary: Gets all aliases from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of aliases.

POST /api/nsk/v1/users/{userKey}/person/aliases

Summary: Creates a new alias on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey}

Summary: Gets a specific alias from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAliasKey	path	string	Yes	The unique person alias key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey}

Summary: Updates a specific alias on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The modified alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey}

Summary: Patches a specific alias on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAliasKey	path	string	Yes	The unique person alias key.

Request Body

The patched alias.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/aliases/{personAliasKey}

Summary: Deletes a specific alias on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personAliasKey	path	string	Yes	The unique person alias key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/comments

Summary: Gets all comments from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of comments.

POST /api/nsk/v1/users/{userKey}/person/comments

Summary: Creates a new comment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey}

Summary: Gets a specific comment from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personCommentKey	path	string	Yes	The unique person comment key.

Responses

404 Response

200 Response

The comment.

PUT /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey}

Summary: Updates a specific comment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The modified comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey}

Summary: Patches a specific comment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personCommentKey	path	string	Yes	The unique person comment key.

Request Body

The patched comment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/comments/{personCommentKey}

Summary: Deletes a specific comment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personCommentKey	path	string	Yes	The unique person comment key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/emails

Summary: Gets all email addresses from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	No description

Responses

404 Response

200 Response

The collection of email addresses.

POST /api/nsk/v1/users/{userKey}/person/emails

Summary: Creates a new email address on the specific user's person.

Description

The "type" property for the request is a single character. See the "/resources/EmailTypes" endpoint for valid email type codes.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey}

Summary: Gets a specific email address from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Responses

404 Response

200 Response

The email address.

PUT /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey}

Summary: Updates a specific email address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The specific user key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The modified email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey}

Summary: Patches a specific email address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Request Body

The patched email address.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/emails/{personEmailAddressKey}

Summary: Deletes a specific email address on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personEmailAddressKey	path	string	Yes	The unique person email address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/information

Summary: Gets all information from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of information.

POST /api/nsk/v1/users/{userKey}/person/information

Summary: Creates a new information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/information/{personInformationKey}

Summary: Gets a specific information from the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personInformationKey	path	string	Yes	The unique person information key.

Responses

404 Response

200 Response

The information.

PUT /api/nsk/v1/users/{userKey}/person/information/{personInformationKey}

Summary: Updates a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The modified information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/information/{personInformationKey}

Summary: Patches a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personInformationKey	path	string	Yes	The unique person information key.

Request Body

The patched information.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/information/{personInformationKey}

Summary: Deletes a specific information on the logged in user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personInformationKey	path	string	Yes	The unique person information key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/phoneNumbers

Summary: Gets all phone numbers from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

POST /api/nsk/v1/users/{userKey}/person/phoneNumbers

Summary: Creates a new phone number on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey}

Summary: Gets a specific phone number from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey}

Summary: Updates a specific phone number on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The modified phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey}

Summary: Patches a specific phone number on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Request Body

The patched phone number.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/phoneNumbers/{personPhoneNumberKey}

Summary: Deletes a specific phone number on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPhoneNumberKey	path	string	Yes	The unique person phone number key.

Responses

200 Response

GET /api/nsk/v1/users/{userKey}/person/preferences

Summary: Gets all preferences for a specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique person key.

Responses

404 Response

200 Response

The collection of preferences.

POST /api/nsk/v1/users/{userKey}/person/preferences

Summary: Creates a new preference for a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey}

Summary: Gets a specific preference for a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

404 Response

200 Response

The preference.

PUT /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey}

Summary: Updates a specific preference for a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The modified preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey}

Summary: Patches a specific preference for a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Request Body

The patched preference.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/preferences/{personPreferenceKey}

Summary: Deletes a specific preference for a specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personPreferenceKey	path	string	Yes	The unique person preference key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/programs

Summary: Gets all programs from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of programs.

POST /api/nsk/v1/users/{userKey}/person/programs

Summary: Creates a new program on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey}

Summary: Gets a specific program from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personProgramKey	path	string	Yes	The unique person program key.

Responses

404 Response

200 Response

The program.

PUT /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey}

Summary: Updates a specific program on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The modified program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey}

Summary: Patches a specific program on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personProgramKey	path	string	Yes	The unique person program key.

Request Body

The patched program.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/programs/{personProgramKey}

Summary: Deletes a specific program on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personProgramKey	path	string	Yes	The unique person address key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/storedPayments

Summary: Gets all stored payments from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of stored payments.

POST /api/nsk/v1/users/{userKey}/person/storedPayments

Summary: Creates a new stored payment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey}

Summary: Gets a specific stored payment from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Responses

404 Response

200 Response

The stored payment.

PUT /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey}

Summary: Updates a specific stored payment on the specific user's person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The modified stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey}

Summary: Patches a specific stored payment on the specific user's person.

Description

To update the account number please DELETE and POST a new stored payment.

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Request Body

The patched stored payment.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/storedPayments/{personStoredPaymentKey}

Summary: Deletes a specific stored payment on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personStoredPaymentKey	path	string	Yes	The unique person stored payment key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/{userKey}/person/travelDocuments

Summary: Gets all travel documents from the specific user's person.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of travel documents.

POST /api/nsk/v1/users/{userKey}/person/travelDocuments

Summary: Creates a new travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/person/travelDocuments/{personTravelDocumentKey}

Summary: Gets a specific travel document from the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Responses

404 Response

200 Response

The travel document.

DELETE /api/nsk/v1/users/{userKey}/person/travelDocuments/{personTravelDocumentKey}

Summary: Deletes a specific travel document on the specific user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Responses

200 Response

Ok.

PUT /api/nsk/v2/users/{userKey}/person/travelDocuments/{personTravelDocumentKey}

Summary: Updates a specific travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The modified travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v2/users/{userKey}/person/travelDocuments/{personTravelDocumentKey}

Summary: Patches a specific travel document on the specific user's person. Note: Gender and Date of Birth on the Travel Document will overwrite the values on the related person object. This is by design due to New Skies validation on the Gender property, and on the Travel Document Date of Birth being trusted over the person record Date of Birth.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
personTravelDocumentKey	path	string	Yes	The unique person travel document key.

Request Body

The patched travel document.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

GET /api/nsk/v1/users/{userKey}/person/travelNotifications

Summary: Gets person travel notifications for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.

Responses

404 Response

200 Response

POST /api/nsk/v1/users/{userKey}/person/travelNotifications

Summary: Creates a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.

Request Body

The travel notification create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}

Summary: Gets a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The person travel notification key.

Responses

404 Response

200 Response

PUT /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}

Summary: Updates a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

PATCH /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}

Summary: Patches a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The delta mapper travel notification edit request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}

Summary: Deletes a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

GET /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events

Summary: Gets notification events for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST /api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events

Summary: Creates a notification event for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification event create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Gets a notification event for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The event type. Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation

Responses

200 Response

404 Response

DELETE

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/events/{eventType}

Summary: Deletes a notification event from a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.
eventType	path	string	Yes	The notification event type.
Enumeration values: 0 = DepartureDelay, 1 = ArrivalDelay, 2 = ScheduleChange, 3 = CheckIn, 4 = GateInformation				

Responses

200 Response

GET

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Gets notification timed events for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Responses

200 Response

404 Response

POST

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents

Summary: Creates a notification timed event for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.

Request Body

The notification timed even create request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventId}

Summary: Gets a notification timed event for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

404 Response

PUT

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timedEventId}

Summary: Updates a notification timed event for a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Request Body

The updated notification timed event request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

500 Response

DELETE

/api/nsk/v1/users/{userKey}/person/travelNotifications/{travelNotificationKey}/timedEvents/{timed}

Summary: Deletes a notification timed event from a person travel notification for a user's person.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The user key.
travelNotificationKey	path	string	Yes	The travel notification key.
timedEventType	path	string	Yes	The notification timed event type.
Enumeration values: 0 = Departure, 1 = Arrival				

Responses

200 Response

GET /api/nsk/v2/users/{userKey}/preferences/skySpeedSettings

Summary: Retrieves the default SkySpeed settings user preferences of the specified user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique identifier of the user.

Responses

200 Response

404 Response

Examples

Response 200

Sky Speed preferences found.:

```
json

{
  "data": {
    "currencyCode": "USD",
    "cultureCode": "en-US",
    "countryCode": "US",
    "nationality": "US",
    "collectionCurrencyCode": "EUR",
    "collectionCurrencySetting": 2,
    "flightResultFareType": 1
  }
}
```

Description: Example response when Sky Speed preferences are successfully retrieved.

Response 404

Sky Speed preferences not found:

```
json

{
  "data": null
}
```

Description: Example response when a GET returns a 404 not found HTTP status code.

PUT /api/nsk/v2/users/{userKey}/preferences/skySpeedSettings

Summary: Updates the default SkySpeed settings user preferences of the specified user.

Description

If the SkySpeed settings of the current user is disabled and an update is made, the settings will automatically be enabled and updated.

If given, each string property is validated if it exists and is active.

The collection currency setting can only be CollectionCurrencySetting.None if the collection currency code is null or empty.

A bad request will be returned if the collection currency code and collection currency setting are:

```
plaintext

- Collection Currency Code is not null or empty and the Collection Currency Setting is set to
CollectionCurrency.None.
- Collection Currency Code is null or empty and the Collection Currency Setting is set to
CollectionCurrency.Both (Any value except for CollectionCurrencySetting.None).
```

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique identifier of the user.

Request Body

The SkySpeed settings user preferences.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Request

Update Sky Speed settings:

```
json   
{  
    "currencyCode": "USD",  
    "cultureCode": "en-US",  
    "countryCode": "US",  
    "nationality": "US",  
    "collectionCurrencyCode": "EUR",  
    "collectionCurrencySetting": 3,  
    "flightResultFareType": 0  
}
```

Description: Example request for updating Sky Speed user settings preferences.

Response 200

Sky Speed preferences updated:

```
json   
{  
    "data": null  
}
```

Description: Example response when update is successful.

Response 400

String length attribute errors:



json



```
{
  "errors": [
    {
      "id": null,
      "code": "SkySpeedSettingsUserPreferences.CurrencyCode:StringLengthAttribute",
      "message": "SkySpeedSettingsUserPreferences.CurrencyCode:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "member": "CurrencyCode",
        "model": "SkySpeedSettingsUserPreferences",
        "validation": "StringLengthAttribute"
      },
      "rawMessage": "The field SkySpeedSettingsUserPreferences must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "SkySpeedSettingsUserPreferences.CultureCode:StringLengthAttribute",
      "message": "SkySpeedSettingsUserPreferences.CultureCode:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "member": "CultureCode",
        "model": "SkySpeedSettingsUserPreferences",
        "validation": "StringLengthAttribute"
      },
      "rawMessage": "The field SkySpeedSettingsUserPreferences must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "SkySpeedSettingsUserPreferences.CountryCode:StringLengthAttribute",
      "message": "SkySpeedSettingsUserPreferences.CountryCode:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "member": "CountryCode",
        "model": "SkySpeedSettingsUserPreferences",
        "validation": "StringLengthAttribute"
      },
      "rawMessage": "The field SkySpeedSettingsUserPreferences must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "SkySpeedSettingsUserPreferences.Nationality:StringLengthAttribute",
      "message": "SkySpeedSettingsUserPreferences.Nationality:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "member": "Nationality",
        "model": "SkySpeedSettingsUserPreferences",
        "validation": "StringLengthAttribute"
      },
      "rawMessage": "The field SkySpeedSettingsUserPreferences must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "SkySpeedSettingsUserPreferences.CollectionCurrencyCode:StringLengthAttribute",
      "message": "SkySpeedSettingsUserPreferences.CollectionCurrencyCode:StringLengthAttribute",
      "type": "Validation",
      "details": {
        "member": "CollectionCurrencyCode",
        "model": "SkySpeedSettingsUserPreferences",
        "validation": "StringLengthAttribute"
      },
      "rawMessage": "The field SkySpeedSettingsUserPreferences must be a string with a maximum length of 3.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ]
}
```

```
        }
    ],
    "data": null
}
```

Description: Example response when a request property string length is too long.

Invalid request codes:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidCurrencyCode",
      "message": "The currency code `AAA` does not exist or is inactive.",
      "type": null,
      "details": {
        "currencyCode": "AAA"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidCultureCode",
      "message": "The culture code `AS-ASD` does not exist or is inactive.",
      "type": null,
      "details": {
        "cultureCode": "AS-ASD"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidCountryCode",
      "message": "The country code `QQ` does not exist or is inactive.",
      "type": null,
      "details": {
        "countryCode": "QQ"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidNationality",
      "message": "The nationality `QQ` does not exist or is inactive.",
      "type": null,
      "details": {
        "nationality": "QQ"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    },
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidCollectionCurrencyCode",
      "message": "The collection currency code `AAA` does not exist or is inactive.",
      "type": null,
      "details": {
        "collectionCurrencyCode": "AAA"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when certain codes have invalid data.

Enum out of range error code:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "core:Validation:EnumOutOfRange",
      "message": "core:Validation:EnumOutOfRange",
      "type": "Validation",
      "details": {
        "name": "CollectionCurrencySetting",
        "value": "6"
      },
      "rawMessage": "The enum 'CollectionCurrencySetting' contains an invalid value of '6'.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when an enumeration request value is out of the expected enumeration values.

Invalid collected currency code and setting combination::

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:InvalidCollectionCurrencyCodeAndSetting",
      "message": "nsk:UserPreferences:SkySpeedSettings:InvalidCollectionCurrencyCodeAndSetting",
      "type": null,
      "details": {
        "collectionCurrencyCode": "USD",
        "collectionCurrencyCodeSetting": "None"
      },
      "rawMessage": "The collection currency code `USD` and setting `None` are invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when an invalid combination of collected currency code and setting is requested.

Null request::

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:UserPreferences:SkySpeedSettings:SkySpeedSettingsUserPreferencesRequired",
      "message": "nsk:UserPreferences:SkySpeedSettings:SkySpeedSettingsUserPreferencesRequired",
      "type": "Error",
      "details": null,
      "rawMessage": "The SkySpeedSettingsUserPreferences is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when null request is received by the endpoint.

GET /api/nsk/v1/users/{userKey}/roles

Summary: Gets all roles for a specific user.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Responses

404 Response

200 Response

The collection of roles.

POST /api/nsk/v1/users/{userKey}/roles

Summary: Creates a new role for a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.

Request Body

The new role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

Created.

GET /api/nsk/v1/users/{userKey}/roles/{userRoleKey}

Summary: Gets a specific role for a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
userRoleKey	path	string	Yes	The unique user role key

Responses

404 Response

200 Response

The user role.

PUT /api/nsk/v1/users/{userKey}/roles/{userRoleKey}

Summary: Updates a specific role for a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
userRoleKey	path	string	Yes	The unique user role key.

Request Body

The modified role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

PATCH /api/nsk/v1/users/{userKey}/roles/{userRoleKey}

Summary: Patches a specific role for a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
userRoleKey	path	string	Yes	The unique user role key.

Request Body

The patched role.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

Ok.

500 Response

DELETE /api/nsk/v1/users/{userKey}/roles/{userRoleKey}

Summary: Deletes a specific role for a specific user.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
userKey	path	string	Yes	The unique user key.
userRoleKey	path	string	Yes	The unique user role key.

Responses

200 Response

Ok.

GET /api/nsk/v1/users/byPerson/{personKey}

Summary: Retrieves the a specific user by the person key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
personKey	path	string	Yes	The person key of the user to retrieve.

Responses

404 Response

The item was not found.

200 Response

OK. Request completed successfully.

Utilities Endpoints

GET /api/nsk/v1/utilities/stationLocalTime/{stationCode}

Summary: Retrieves the station local time.

Description

No description available

Request Details

Parameters

Name	Location	Type	Required	Description
stationCode	path	string	Yes	The station code to get the local time for.

Responses

404 Response

200 Response

Verifiedtraveldocuments Endpoints

PUT /api/dcs/v1/verifiedTravelDocuments

Summary: Updates a list of verified travel documents for a passenger and associated infant.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext



The following items are required to use this endpoint.

Agent permissions.

Government Security Drivers need to be enabled under government security configurations.

Government Security Drivers need to be configured to approve the given document type.

The given document must be a document type approved by customs.

The target passenger must be either checked in or boarded on the target segment.

Due to the nature of this endpoint there is no data returned. To confirm successful update you can find an entry in the booking history using `/api/nsk/v1/bookings/{bookingKey}/history` or view passenger segment information by pulling the booking using `/api/nsk/v1/bookings/{recordLocator}`

If you would like to remove all verified travel documents from a passenger please use DELETE `/api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey}`

Request Details

Request Body

The request for updating verified travel documents on a bookings.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Request

Example updated verified travel documents request:

json



```
{  
  "recordLocator": "ABC123",  
  "segmentKey": "T1Z_NTYwOH4gfn5TTEN_MDcvMjEvMjAyNCAxNzowMH5CT1N_MDcvMjEvMjAyNCAyMzo10X5_",  
  "passengerKey": "MCFBRFQ-",  
  "verifiedTravelDocuments": [  
    {  
      "documentTypeCode": "P",  
      "issuedByCode": "US",  
      "isInfant": false  
    }  
  ]  
}
```

Description: An example of an update verified travel documents request.

Response 400

Invalid Record Locator Error:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "recordLocator": "9SkFNRegdQRYuQ3YfA2g9"  
      },  
      "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3YfA2g9' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
  ],  
  "data": null  
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "passengerKey": "JoKwkYODu4gNYRwNQo"  
      },  
      "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
  ],  
  "data": null  
}
```

Description: Example error when passenger key is invalid.

Invalid Segment Key Error:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "segmentKey": "JoKwkYODu4gNYRwNQo"  
      },  
      "rawMessage": "The identifier 'SegmentKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    },  
  ],  
  "data": null  
}
```

Description: Example error when segment key is invalid.

Invalid Lift Status Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidLiftStatus",
      "message": "Updating verified travel documents requires the passenger to have a lift status of checked in or boarded. Lift s",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r"
    }
  ],
  "data": null
}
```

Description: Example error when lift status is invalid.

Invalid Document Type Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidDocumentType",
      "message": "The document type code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r"
    }
  ],
  "data": null
}
```

Description: Example error when document type is invalid.

Invalid Issued by Code Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidIssuedByCode",
      "message": "The issued by code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r"
    }
  ],
  "data": null
}
```

Description: Example error when document issued by code is invalid.

POST

/api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passen

Summary: Adds a new verified travel document to a passenger or the associated infant.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

plaintext



The following items are required to use this endpoint.

Agent permissions.

Government Security Drivers need to be enabled under government security configurations.

Government Security Drivers need to be configured to approve the given document type.

The given document must be a document type approved by customs.

The target passenger must be either checked in or boarded on the target segment.

Due to the nature of this endpoint there is no data returned. To confirm successful update you can find an entry in the booking history using GET `/api/nsk/v1/bookings/{bookingKey}/history` or view passenger segment information by pulling the booking using GET `/api/nsk/v1/bookings/{recordLocator}`

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the target booking.
segmentKey	path	string	Yes	The target segment key.
passengerKey	path	string	Yes	The target passenger key.

Request Body

The request for updating verified travel documents on a bookings.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

400 Response

Examples

Request

Example verified travel document:

json



```
{  
  "documentTypeCode": "P",  
  "issuedByCode": "US",  
  "isInfant": false  
}
```

Description: An example of a basic verified travel document.

Response 400

Invalid Record Locator Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "recordLocator": "9SkFNRegdQRYuQ3YfA2g9"
      },
      "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3YfA2g9' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "passengerKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when passenger key is invalid.

Invalid Segment Key Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "segmentKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'SegmentKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when segment key is invalid.

Invalid Lift Status Error:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidLiftStatus",
      "message": "Updating verified travel documents requires the passenger to have a lift status of checked in or boarded. Lift s",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}

```

Description: Example error when lift status is invalid.

Invalid Document Type Error:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidDocumentType",
      "message": "The document type code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}

```

Description: Example error when document type is invalid.

Invalid Issued by Code Error:

```

json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidIssuedByCode",
      "message": "The issued by code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}

```

Description: Example error when document issued by code is invalid.

DELETE

[`/api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passen`](#)

Summary: Deletes the verified travel documents for a passenger and associated infant.

Description

Requires agent permissions.

```
plaintext   
  
This will delete all verified documents associated with the passenger which includes infant travel documents.  
If you would like to update the list without removing all of them please use PUT  
`/api/dcs/v1/verifiedTravelDocuments`  
  
IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.
```

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the target booking.
segmentKey	path	string	Yes	The target segment key.
passengerKey	path	string	Yes	The target passenger key.

Responses

200 Response

400 Response

Examples

Response 400

Invalid Record Locator Error:

```
json   
  
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "recordLocator": "9SkFNRegdQRYuQ3YfA2g9"  
      },  
      "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3YfA2g9' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    },  
    {  
      "data": null  
    }  
  ]  
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "passengerKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when passenger key is invalid.

Invalid Segment Key Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "segmentKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'SegmentKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when segment key is invalid.

Invalid Lift Status Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidLiftStatus",
      "message": "Updating verified travel documents requires the passenger to have a lift status of checked in or boarded. Lift status must be present in the request body.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when lift status is invalid.

Invalid Document Type Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidDocumentType",
      "message": "The document type code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when document type is invalid.

Invalid Issued by Code Error:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidIssuedByCode",
      "message": "The issued by code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."
    }
  ],
  "data": null
}
```

Description: Example error when document issued by code is invalid.

POST

[/api/dcs/v1/verifiedTravelDocuments/{recordLocator}/segments/{segmentKey}/passengers/{passengerKey}/documents](#)

Summary: Adds a new verified travel document to a passenger or the associated infant by the document key.

Description

IMPORTANT: This endpoint should NOT be called concurrently using the same session token to avoid performance problems.

```
plaintext
The following items are required to use this endpoint.

Agent permissions.

Government Security Drivers need to be enabled under government security configurations.

Government Security Drivers need to be configured to approve the given document type.

The given document must be a document type approved by customs.

The target passenger must be either checked in or boarded on the target segment.

Due to the nature of this endpoint there is no data returned. To confirm successful update you can find an entry in the booking history using `/api/nsk/v1/bookings/{bookingKey}/history` or view passenger segment information by pulling the booking using `/api/nsk/v1/bookings/{recordLocator}`
```

Request Details

Parameters

Name	Location	Type	Required	Description
recordLocator	path	string	Yes	The record locator of the target booking.
segmentKey	path	string	Yes	The target segment key.
passengerKey	path	string	Yes	The target passenger key.
documentKey	path	string	Yes	The document key of the document you would like to verify.

Responses

201 Response

400 Response

Examples

Response 400

Invalid Record Locator Error:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "recordLocator": "9SkFNRegdQRYuQ3YfA2g9"  
      },  
      "rawMessage": "The identifier 'RecordLocator' with value '9SkFNRegdQRYuQ3YfA2g9' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ],  
  "data": null  
}
```

Description: Example error when record locator is invalid.

Invalid Passenger Key Error:

```
json   
{  
  "errors": [  
    {  
      "id": null,  
      "code": "nsk:Exceptions:InvalidKey",  
      "message": "nsk:Exceptions:InvalidKey",  
      "type": "Validation",  
      "details": {  
        "passengerKey": "JoKwkYODu4gNYRwNQo"  
      },  
      "rawMessage": "The identifier 'PassengerKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned."  
    }  
  ],  
  "data": null  
}
```

Description: Example error when passenger key is invalid.

Invalid Segment Key Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk:Exceptions:InvalidKey",
      "message": "nsk:Exceptions:InvalidKey",
      "type": "Validation",
      "details": {
        "segmentKey": "JoKwkYODu4gNYRwNQo"
      },
      "rawMessage": "The identifier 'SegmentKey' with value 'JoKwkYODu4gNYRwNQo' is invalid.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when segment key is invalid.

Invalid Lift Status Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidLiftStatus",
      "message": "Updating verified travel documents requires the passenger to have a lift status of checked in or boarded. Lift status is required for travel document updates.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when lift status is invalid.

Invalid Document Type Error:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidDocumentType",
      "message": "The document type code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example error when document type is invalid.

Invalid Issued by Code Error:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "dcs:VerifiedTravelDocuments:InvalidIssuedByCode",
      "message": "The issued by code 'INVALID' is invalid or inactive.",
      "type": "Validation",
      "details": null,
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example error when document issued by code is invalid.

Voucherissuance Endpoints

POST /api/nsk/v1/voucherIssuance

Summary: Creates vouchers based upon data in the request.

Description

This endpoint requires agent permissions. Note that creating high volume voucher issuance results in a batch job which may not process immediately.

Request Details

Request Body

The voucher issuance request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

201 Response

GET /api/nsk/v1/voucherIssuance/{voucherIssuanceKey}

Summary: Gets a voucher issuance based upon the voucher issuance key.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherIssuanceKey	path	string	Yes	The voucher issuance key.

Responses

200 Response

404 Response

Vouchers Endpoints

GET /api/nsk/v2/vouchers

Summary: Gets vouchers based upon data in the request.

Description

This endpoint requires agent permissions. If given a date, one of the following properties are required:

plaintext	copy
- CustomerName.LastName	
- CustomerNumber	
- Agent	
- Agent.Name	
- Agent.Domain	

If only the begin date is given, the default value of the end date will be the day after the begin date. If only the end date is given, the begin date will be required.

If at least one of the market properties is provided, all the other market properties are required. If at least one of the agent properties is provided, the other agent property is required. If the customer first name is provided, the customer last name is required.

Request Details

Parameters

Name	Location	Type	Required	Description
VoucherIssuanceKey	query	string	No	The unique voucher issuance key.

plaintext	copy

| Market.Destination | query | string | No | Gets or sets the leg arrival station. || Market.Origin | query | string | No | Gets or sets the leg departure station. || Market.DepartureDate | query | string (date-time) | No | The departure date. || Market.Identifier | query | string | No | The unique transportation identifier. || Market.CarrierCode | query | string | No | The carrier code. || Market.OpSuffix | query | string | No | The op suffix. Please note that this should be a char and not a string. || Agent.Name | query | string | No | The agent name associated with the voucher. || Agent.Domain | query | string | No | The domain code associated with the voucher. || BeginDate | query | string (date-time) | No | The starting date. || EndDate | query | string (date-time) | No | The ending date. || PageSize | query | integer | No | The Page Size, Default 1000 || LastPageKey | query | string | No | The LastPageKey, where to start paged query. || SortCriteria | query | string | No | Defines the criteria in which the find vouchers result can be sorted. Defaults to `CreatedDateAscending` if not specified or invalid.

Enumeration values: 0 = `CreatedDateAscending`, 1 = `Name`, 2 = `CreatedDateDescending` || RecordLocator | query | string | No | The record locator associated with the voucher. || CustomerName.FirstName | query | string | No | The first name associated with the voucher. || CustomerName.LastName | query | string | No | The last name associated with the voucher. || CustomerNumber | query | string | No | The customer number associated with the voucher. || ActiveOnly | query | boolean | No | The flag to indicate if only active items should be returned. || CultureCode | query | string | No | The culture code of the paged request. |

Responses

200 Response

404 Response

400 Response

Examples

Response 400

No filters specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:VoucherNotFound",
      "message": "nsk-server:VoucherNotFound",
      "type": "Validation",
      "details": null,
      "rawMessage": "Please specify any of the search criteria: Voucher Details/Passenger Details/Record Locator/Flight Details/Agreement Details",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Example response when no filters are specified.

At least one of the market properties is specified and no destination is specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformation.Destination:RequiredAttribute",
      "message": "VoucherMarketInformation.Destination:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformation",
        "member": "Destination"
      },
      "rawMessage": "The VoucherMarketInformation field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be returned"
    }
  ],
  "data": null
}
```

Description: Example response when at least one of the market properties is specified and the destination is set to null or an empty string.

At least one of the market properties is specified and no origin is specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformation.Origin:RequiredAttribute",
      "message": "VoucherMarketInformation.Origin:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformation",
        "member": "Origin"
      },
      "rawMessage": "The VoucherMarketInformation field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be r
    }
  ],
  "data": null
}
```

Description: Example response when at least one of the market properties is specified and the origin is set to null or an empty string.

At least one of the market properties is specified and no departure date is specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformation.DepartureDate:RequiredAttribute",
      "message": "VoucherMarketInformation.DepartureDate:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformation",
        "member": "DepartureDate"
      },
      "rawMessage": "The VoucherMarketInformation field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when at least one of the market properties is specified and the departure date is set to null.

At least one of the market properties is specified and no identifier is specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformation.Identifier:RequiredAttribute",
      "message": "VoucherMarketInformation.Identifier:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformation",
        "member": "Identifier"
      },
      "rawMessage": "The VoucherMarketInformation field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when at least one of the market properties is specified and the identifier is set to null or an empty string.

At least one of the market properties is specified and no carrier code is specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformation.CarrierCode:RequiredAttribute",
      "message": "VoucherMarketInformation.CarrierCode:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformation",
        "member": "CarrierCode"
      },
      "rawMessage": "The VoucherMarketInformation field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when at least one of the market properties is specified and the carrier code is set to null or an empty string.

Agent name is specified and agent domain is not specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "Agent.Domain:RequiredAttribute",
      "message": "Agent.Domain:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "Agent",
        "member": "Domain"
      },
      "rawMessage": "The Agent field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the agent name is specified and the agent domain is set to null or an empty string.

Agent domain is specified and agent name is not specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "Agent.Name:RequiredAttribute",
      "message": "Agent.Name:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "Agent",
        "member": "Name"
      },
      "rawMessage": "The Agent field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the agent domain is specified and the agent name is set to null or an empty string.

Customer first name is specified and customer last name is not specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "CustomerName.LastName:RequiredAttribute",
      "message": "CustomerName.LastName:RequiredAttribute",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "CustomerName",
        "member": "LastName"
      },
      "rawMessage": "The CustomerName field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the customer first name is specified and the customer last name is set to null or an empty string.

End date is specified and begin date is not specified:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Voucher:VoucherBeginDateRequired",
      "message": "nsk:Voucher:VoucherBeginDateRequired",
      "type": "Validation",
      "details": {
        "beginDate": null
      },
      "rawMessage": "Voucher creation begin date is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when end date is specified and begin date is null.

End date is earlier than begin date:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:EndDate:EndDateBeforeBeginDate",
      "message": "nsk:EndDate:EndDateBeforeBeginDate",
      "type": "Validation",
      "details": null,
      "rawMessage": "The requested end date of 4/29/2025 occurs before the requested begin date of 4/30/2025.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the end date is earlier than the begin date.

Begin date or end date is specified but customer name, customer number, agent name, and agent domain is not specified:

```
json
{
  "errors": [
    {
      "id": null,
      "code": "nsk:Voucher:VoucherSearchRequirementsUnmet",
      "message": "nsk:Voucher:VoucherSearchRequirementsUnmet",
      "type": "Validation",
      "details": {
        "customerName": null,
        "customerNumber": null,
        "agentName": null,
        "agentDomainCode": null
      },
      "rawMessage": "Requirements unmet: Enter one - Customer Name, Customer Number or Agent Name.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the Begin date or end date is specified but customer name, customer number, agent name, and agent domain is set to null or an empty string.

GET /api/nsk/v1/vouchers/{voucherKey}

Summary: Gets a specific voucher based upon the voucher key.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherKey	path	string	Yes	The voucher key.

Responses

200 Response

404 Response

PATCH /api/nsk/v1/vouchers/{voucherKey}

Summary: Changes the voucher status, type, or expiration date.

Description

This endpoint requires agent permissions. Due to system limitations only one field can be updated per call. Voucher types of 0 and 2 (Credit and Service) have been removed from the request for this endpoint as they are not valid type to be updated to.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherKey	path	string	Yes	The voucher key

Request Body

The voucher update request.

Content Type: application/json

Schema: See API specification for detailed schema

Responses

200 Response

400 Response

Examples

Response 400

Multiple Field Update Error:

json



```
{
  "errors": [
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing VoucherType when changing the Status on a Voucher to Void is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing VoucherType when changing the Status on a Voucher to Void is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    },
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing ExpirationDate when changing the Status on a Voucher to Void is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing ExpirationDate when changing the Status on a Voucher to Void is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    },
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing VoucherType when reinstating a Voucher is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing VoucherType when reinstating a Voucher is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    },
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing ExpirationDate when reinstating a Voucher is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing ExpirationDate when reinstating a Voucher is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    },
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing the Status of a Voucher when extending the ExpirationDate is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing the Status of a Voucher when extending the ExpirationDate is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    },
    {
      "id": null,
      "code": "nsk-server:Vouchers",
      "message": "Changing the VoucherType of a Voucher when extending the ExpirationDate is not supported",
      "type": "Error",
      "details": {
        "voucherWarningMessage": "Changing the VoucherType of a Voucher when extending the ExpirationDate is not supported"
      },
      "rawMessage": null,
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be rendered"
    }
  ]
}
```

```

    "exception": null,
    "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present",
},
{
    "id": null,
    "code": "nsk-server:Vouchers",
    "message": "Changing the Status of a Voucher when changing the VoucherType is not supported",
    "type": "Error",
    "details": {
        "voucherWarningMessage": "Changing the Status of a Voucher when changing the VoucherType is not supported"
    },
    "rawMessage": null,
    "exception": null,
    "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present",
},
{
    "id": null,
    "code": "nsk-server:Vouchers",
    "message": "Changing ExpirationDate of a Voucher when changing a VoucherType is not supported",
    "type": "Error",
    "details": {
        "voucherWarningMessage": "Changing ExpirationDate of a Voucher when changing a VoucherType is not supported"
    },
    "rawMessage": null,
    "exception": null,
    "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present"
},
],
"data": null
}

```

Description: Example error if more than one field is being updated

Invalid Voucher Type Error:

```

json
{
    "errors": [
        {
            "id": null,
            "code": "nsk:Exceptions:OutOfRange",
            "message": "The value '2' is out of range for 'VoucherUpdateRequestv2.Type'.Expected values: (1,3,4,5,6)",
            "type": "Validation",
            "details": null,
            "rawMessage": null,
            "exception": null,
            "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present"
        }
    ],
    "data": null
}

```

Description: Example error if an invalid voucher type is used

GET /api/nsk/v2/vouchers/byAgent

Summary: Gets a collection of vouchers based on the specified agent information.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
Name	query	string	No	The agent name associated with the voucher.

plaintext



| Domain | query | string | No | The domain code associated with the voucher. || BeginDate | query | string (date-time) | No | The starting date. || EndDate | query | string (date-time) | No | The ending date. || PageSize | query | integer | No | The Page Size, Default 1000 || LastPageKey | query | string | No | The LastPageKey, where to start paged query. || SortCriteria | query | string | No | Defines the criteria in which the find vouchers result can be sorted. Defaults to `CreatedDateAscending` if not specified or invalid.

Enumeration values: 0 = `CreatedDateAscending`, 1 = `Name`, 2 = `CreatedDateDescending` |

Responses

200 Response

404 Response

GET /api/nsk/v1/vouchers/byDate

Summary: Searches for vouchers issued within the given date range.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
PageSize	query	integer	No	The number of items to return.

plaintext



| LastPageKey | query | string | No | The key returned previously to get the next page of items. || BeginDate | query | string (date-time) | No | The starting date. || EndDate | query | string (date-time) | No | The ending date. |

Responses

200 Response

404 Response

GET /api/nsk/v2/vouchers/byIssuance/{voucherIssuanceKey}

Summary: Gets vouchers associated with a voucher issuance.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherIssuanceKey	path	string	Yes	The voucher issuance key.
PageSize	query	integer	No	The Page Size, Default 1000

plaintext



| LastPageKey | query | string | No | The LastPageKey, where to start paged query. || sortCriteria | query | string | No | Indicates how the search results will be sorted. Defaults to `CreatedDate` if not specified or invalid.

Enumeration values: 0 = `CreatedDateAscending`, 1 = `Name`, 2 = `CreatedDateDescending` |

Responses

200 Response

OK. Request completed successfully.

404 Response

The item was not found.

400 Response

There was a problem with the request and it could not be completed.

Examples

Response 200

Retrieve Vouchers:

```

json
{
  "data": [
    {
      "lastPageKey": "MTAwMA--",
      "vouchers": [
        {
          "reference": "1001170704130001",
          "configurationCode": "NOREQ",
          "status": 0,
          "available": 1000.0,
          "firstName": "John",
          "lastName": "Doe",
          "personKey": "MUNzPRfx",
          "voucherIssuanceKey": "MTE3MCFHQkFUQ0g-",
          "expiration": "9999-12-31T23:59:59.999999",
          "currencyCode": "USD",
          "amount": 1000.0
        },
        {
          "reference": "1001170704130002",
          "configurationCode": "NOREQ",
          "status": 0,
          "available": 1000.0,
          "firstName": "John",
          "lastName": "Doe",
          "personKey": "MUNzPRfx",
          "voucherIssuanceKey": "MTE3MCFHQkFUQ0g-",
          "expiration": "9999-12-31T23:59:59.999999",
          "currencyCode": "USD",
          "amount": 1000.0
        },
        {
          "reference": "1001170704130003",
          "configurationCode": "NOREQ",
          "status": 0,
          "available": 1000.0,
          "firstName": "John",
          "lastName": "Doe",
          "personKey": "MUNzPRfx",
          "voucherIssuanceKey": "MTE3MCFHQkFUQ0g-",
          "expiration": "9999-12-31T23:59:59.999999",
          "currencyCode": "USD",
          "amount": 1000.0
        }
      ]
    }
  ]
}

```

Description: Successful retrieval of vouchers

Invalid Sort Criteria:

```

json

{
  "messages": [
    {
      "code": "core:Validation:MalformedRequest",
      "type": "Information",
      "value": "The value 'CreatedTime' is invalid.",
      "status": 1,
      "details": null
    }
  ],
  "data": {
    "lastPageKey": "MTAwMA--",
    "vouchers": [
      {
        "createdDate": "2025-08-07T04:47:16.9363258+00:00",
        "voucherKey": "NTA2MiExMTcwITEwMDExNzA3MDQzMzAwMDAx",
        "reference": "1001170704130001",
        "configurationCode": "NOREQ",
        "status": 0,
        "available": 1000.0,
        "firstName": "John",
        "lastName": "Doe",
        "personKey": "MUNzPRfx",
        "voucherIssuanceKey": "MTE3MCFHQkFUQ0g-",
        "expiration": "9999-12-31T23:59:59.999999",
        "currencyCode": "USD",
        "amount": 1000.0
      }
    ]
  }
}

```

Description: The provided sort criteria is invalid or not defined in the system.

GET /api/nsk/v2/vouchers/byMarket

Summary: Gets a collection of vouchers based on the market information.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
Destination	query	string	No	Gets or sets the leg arrival station.

```

plaintext
| |

```

| Origin | query | string | No | Gets or sets the leg departure station. || DepartureDate | query | string (date-time) | No | The departure date. || Identifier | query | string | No | The unique transportation identifier. || CarrierCode | query | string | No | The carrier code. || OpSuffix | query | string | No | The op suffix. Please note that this should be a char and not a string. || PageSize | query | integer | No | The Page Size, Default 1000 || LastPageKey | query | string | No | The LastPageKey, where to start paged query. || SortCriteria | query | string | No | Defines the criteria in which the find vouchers result can be sorted. Defaults to `CreatedDateAscending` if not specified or invalid.

Enumeration values: 0 = CreatedDateAscending, 1 = Name, 2 = CreatedDateDescending |

Responses

200 Response

404 Response

400 Response

Examples

Response 400

No destination specified:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "VoucherMarketInformationv2.Destination:RequiredAttribute",  
      "message": "The VoucherMarketInformationv2 field is required.",  
      "type": "Validation",  
      "details": {  
        "validation": "RequiredAttribute",  
        "model": "VoucherMarketInformationv2",  
        "member": "Destination"  
      },  
      "rawMessage": "The VoucherMarketInformationv2 field is required.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when the destination is set to null or an empty string.

No origin specified:

```
json 
```

```
{  
  "errors": [  
    {  
      "id": null,  
      "code": "VoucherMarketInformationv2.Origin:RequiredAttribute",  
      "message": "The VoucherMarketInformationv2 field is required.",  
      "type": "Validation",  
      "details": {  
        "validation": "RequiredAttribute",  
        "model": "VoucherMarketInformationv2",  
        "member": "Origin"  
      },  
      "rawMessage": "The VoucherMarketInformationv2 field is required.",  
      "exception": null,  
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."  
    }  
  ],  
  "data": null  
}
```

Description: Example response when the origin is set to null or an empty string.

No departure date specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformationv2.DepartureDate:RequiredAttribute",
      "message": "The VoucherMarketInformationv2 field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformationv2",
        "member": "DepartureDate"
      },
      "rawMessage": "The VoucherMarketInformationv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the departure date is set to null.

No identifier specified:

```
json

{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformationv2.Identifier:RequiredAttribute",
      "message": "The VoucherMarketInformationv2 field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformationv2",
        "member": "Identifier"
      },
      "rawMessage": "The VoucherMarketInformationv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}
```

Description: Example response when the identifier is set to null or an empty string.

No carrier code specified:

```

json
{
  "errors": [
    {
      "id": null,
      "code": "VoucherMarketInformationv2.CarrierCode:RequiredAttribute",
      "message": "The VoucherMarketInformationv2 field is required.",
      "type": "Validation",
      "details": {
        "validation": "RequiredAttribute",
        "model": "VoucherMarketInformationv2",
        "member": "CarrierCode"
      },
      "rawMessage": "The VoucherMarketInformationv2 field is required.",
      "exception": null,
      "debugMode": "The API is currently running in debug mode. In production the 'rawMessage' and 'exception' nodes will not be present."
    }
  ],
  "data": null
}

```

Description: Example response when the carrier code is set to null or an empty string.

GET /api/nsk/v1/vouchers/byReference/{voucherReferenceNumber}

Summary: Gets a specific voucher based upon the voucher reference number.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
voucherReferenceNumber	path	string	Yes	The voucher reference number.

Responses

200 Response

404 Response

GET /api/nsk/v1/vouchers/configuration

Summary: Performs a shallow call to get all available voucher configurations.

Description

This endpoint requires agent permissions. Since this performs a shallow call, some voucher configuration list properties come back empty. To get the complete data, use GET /api/nsk/v1/vouchers/configuration/{configurationCode} which performs a deep call to get a specific voucher configuration based on the configuration code.

Request Details

Responses

200 Response

The collection of voucher configurations.

404 Response

GET /api/nsk/v1/vouchers/configuration/{configurationCode}

Summary: Performs a deep call to get a specific voucher configuration based on the configuration code.

Description

This endpoint requires agent permissions.

Request Details

Parameters

Name	Location	Type	Required	Description
configurationCode	path	string	Yes	The code of the voucher configuration to retrieve.

Responses

200 Response

The voucher configuration.

404 Response

Watchlist Endpoints

GET /api/nsk/v1/watchList/{watchListKey}/persons

Summary: Searches for a collection of watch list persons.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
watchListKey	path	string	Yes	The watch list key.
ReferenceNumber	query	string	No	The watch list reference number.

plaintext

|

Responses

200 Response

404 Response

GET /api/nsk/v1/watchList/persons/{watchListPersonKey}

Summary: Gets a specific watch list person by key.

Description

Request Details

Parameters

Name	Location	Type	Required	Description
watchListPersonKey	path	string	Yes	The watch list person key.

Responses

200 Response

404 Response