Livn API v2.0 Developer Guide

Revision Date: 2023-04-21

1. About Livn and the Livn API v2

Livn is an open connectivity hub that offers real-time access to a large and growing aggregated network of tourism suppliers in the tours & activities space.

Clients who integrate the Livn API range from large GDSs and OTAs to virtually any interested travel retailer. We provide easy access to the global market of tours & activities with real-time availability, rates, as well as curated and enriched content. Clients can make instantly confirmed bookings through a single connection.

Livn operates two distinctly different models in which our clients can connect with suppliers:

- **Livn Direct Connect:** Clients who prefer to build, or even already have existing commercial agreements with suppliers that operate on one of the many reservation systems we have connected, can use the Livn API to unify their various incoming connections into a single powerful API, and to transact with their suppliers with no interference from an intermediary wholesaler.
- **Livn Wholesale:** We have contracted an attractive selection of suppliers of tours & activities from around the world that we are pleased to offer to clients who don't have existing relationships with tour operators and wish to quickly break into this exciting market. As a reseller, clients can instantly open up a new revenue stream for their business, while Livn manages all supplier relationships and payments.

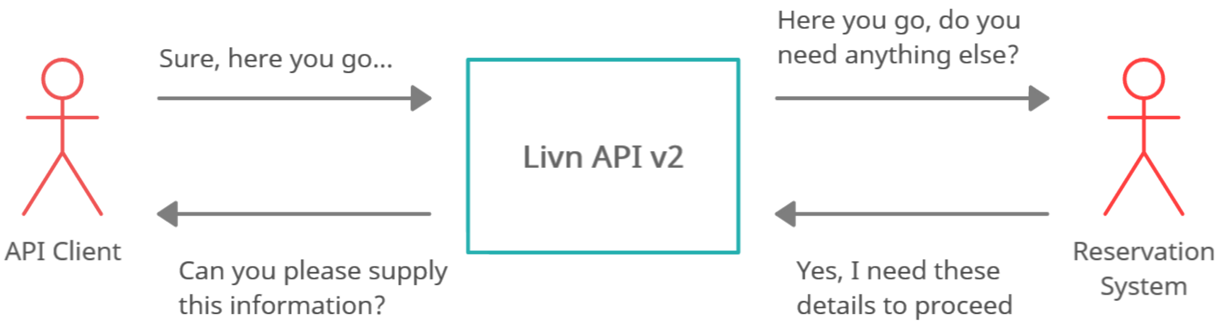
What has proven to be the most popular approach for our customers to engage with Livn is to combine these two models into a hybrid solution, kickstarting their venture into tours & activities by tapping into Livn's extensive wholesale offering and at the same time start building direct relationships with their preferred suppliers. The Livn platform has always been designed to handle both the Direct Connect and Wholesale avenues through the same API, and with v2 we are excited to have brought the two even closer together, allowing clients interested in the Hybrid solution to transact through a single API endpoint.

For more information about Livn please visit www.livn.world

Earlier versions of the Livn API were designed to the limited requirements of a small number of connected reservation systems. But as we started integrating and working with more and more suppliers and new reservation systems, we identified the need for a radically different and highly flexible approach that could cater to the vast degree of complexity and variety present in the data models and processes involved in the distribution of tours & activities products.

After ample deliberation, we realised that the best-suited solution to this challenging problem was the tried and tested way tourism products have traditionally been sold to consumers. This involves breaking up the sales process into two distinct stages:

1. **Product Search & Discovery:** To enable consumers to find their perfect product, our clients require largely uniform and homogenous catalogue-quality content, e.g. product descriptions, images, highlights and includes, etc. This content is complemented by searchable information and metrics such as categories, geolocation data and a well-maintained calendar-style cache of operational dates and lead-in pricing.
2. **Wizard-style Checkout:** Once the customer has taken an interest in a specific product and where applicable, decided on a date to go on the tour or activity of their choice, the actual checkout process has to involve the supplier's respective reservation system. This concrete upstream reservation system will fundamentally dictate the complexity and the number and sequence of steps towards making a confirmed booking. Modelled on an actual in-person sales interaction, the Livn API v2 aims to guide the customer through the checkout in a sequence of logical steps. Every step of this checkout flow aims to provide the customer with all information relevant to the decision-making process in front of them and collects all user input necessary to advance the checkout to the next step by essentially asking questions and validating, transforming and passing the customer's answers and choices through to the upstream reservation system.


Sure, here you go...
Here you go, do you need anything else?
API Client
Livn API v2
Reservation System
Can you please supply this information?
Yes, I need these details to proceed

Possibly the biggest difficulty with API aggregation and any attempt to try and unify more than a few different API models into just one specification is the inevitable degradation of the individual upstream systems' data structures. It is a virtually impossible challenge to achieve without something being lost in translation. This is particularly true with regard to originally machine-readable information, which either lacks a suitable equivalent in the unified model or cannot be transformed 1:1 into another machine-readable format.

The Livn API v2 has consciously been designed with a view to service clients who should find it to be an excellent one-stop-shop for tours & activities products and for operation as part of customer-facing applications, rather than developers hoping to treat it like a single common reservation system.

The data models for the product search and discovery stage outlined above are rich in clearly structured machine-readable data that should be easy to integrate into an existing application drawing on other sources for similar or entirely different tourism verticals and types of products.

However, the ultra-flexible nature of the checkout stage, needed to overcome the combined complexities of dozens of different source reservation systems, is in large part focusing on human consumers, making this admittedly a difficult API for any further aggregation. We strive to preserve as much machine-readable information as possible, taking second place only to safeguarding that all functionality of the reservation systems is maintained and made available to your customers. If a reservation system offers optional add-on products to be booked in conjunction with a tour ticket, or the option to book a hotel pick-up or the ability to make a two-step checkout with a temporary reservation leading to a confirmed booking, the Livn API will always aim to maintain this functionality and make it available to our clients.

2. Definition of Terms

For a successful integration of the Livn API, we believe it is important for developers to have a good and clear understanding of the terminology used by Livn at large, and throughout this guide and other API documentation resources in particular.

You can find a detailed explanation of the entire API schema, i.e. every API operation as well as all data model types and their properties/fields in our OpenAPI specifications and Swagger-UI page: https://dev.livnapi.com/public

Add On

Optional add-on products that can be purchased in conjunction with the main Product.

Answer

The input supplied by the end customer and API client in response to a specific Question.

Base Variant

Base variants are the first level of differentiation of top-level Products into the ultimately bookable product variants (the other being TimeSlots and Fares), e.g. “Bus” vs “Bus & Boat” options of a city sightseeing pass.

Booking

All details of a booking that was made in one of our upstream Reservation Systems. A confirmed Booking (or potentially multiple Bookings) is the desired result of a completed checkout Flow.

Billing Info

Billing data of the checkout Flow including the billing currency, exchange rate, and forex surcharge (if applicable). The *billingCurrency* is the currency in which Livn is billing a non-DC retail Channel's Seller buying from Livn wholesale, or in the case of a Direct Connect sale the currency Livn uses to invoice its transaction fees. The *billingExchangeRate* is the exchange rate between the *billingCurrency* and the *Flow.currency*.

Cancellation

All details of a cancellation, including failed and yet-in-progress cancellation attempts of a previously confirmed Booking.

Cancellation Policy

The Supplier’s policy regarding future customer-initiated cancellations of a confirmed Booking. Where possible, the rules that form the cancellation policy will be captured and published on the Livn API in a machine-readable format, however, not all Reservation Systems can provide suitable data, leaving a human-readable policy published in textual form as the lowest common denominator.

Catalogue

Like its physical counterpart, a Catalogue contains any number of Products from one or multiple Suppliers. Distributors can create multiple Catalogues to break up and organise their inventory as they see fit, e.g. grouped by destinations, geographic regions or types of products. As Suppliers and Products can exist under more than one Catalogue, owned by the same or different Distributors, we maintain Catalogue-specific copies of both, which is used to manage the Distributor or Catalogue-specific data that goes beyond the information that is shared and can be kept in the Supplier and Product records themselves, e.g. contracted commissions or metadata set by the Distributor who owns the Catalogue.

Category

Categories are used to more or less roughly describe the nature of a Product and to group products of a similar type. Products can fall into more than one category.

Channel

A representation of the link and relationship between a Seller and thus API User and a single Distributor, e.g. Livn's Wholesale offering. Through this Channel, the Distributor can give the API User access to any number of their product Catalogues. Clients who exclusively, or alongside buying from a wholesaler such as Livn, themselves operate as a Distributor under the Livn Direct model, dealing and trading directly with Suppliers, will have a specially marked Channel into their own Distributor’s inventory. Products that are available to the API client both from a wholesaler's inventory and as part of their Direct Connect relationship with the Supplier, will automatically be picked from the latter Channel and booked using their own Credential Set for the upstream Reservation System.

Company

A record of the contact and other basic details of a real-world company. In the scope of the Livn API, it is primarily used as part of a Distributor.

Credential Set

Credential Sets are the various API keys, tokens, usernames and passwords securing the APIs of the Reservation Systems used by our Suppliers. Depending on the Reservation System we may only need a single set to access the entire system, or require separate credentials to access the inventory of individual suppliers or in extreme cases subsets of their inventory. Credential sets are needed to import the rich content of suppliers and their products used during a product search, to maintain the cache of availability and pricing and of course to interact with the reservation system and write live bookings during checkout. Direct Connect clients, who use the Livn API to transact with suppliers on their own commercial terms, are generally issued their own credential sets by the supplier or reservation system. Livn requires these credentials to communicate with the reservation system on the client’s behalf. Acting as custodians of such sensitive information we assure our Direct Connect partners that security is our highest priority and all credential sets are encrypted and securely transferred and stored according to industry standards and best practices.

Departure

A date on which a Product is known to operate and any cached from-prices.

Distributor

A Distributor is a Company that has an inventory of Products, which they purchase directly from any number of Suppliers through their respective Reservation System and distribute, i.e. sell via the Livn API. A Distributor can either sell these products directly via a Direct Connect Channel, or act as a wholesaler for third-party resellers, or even do both. Every Distributor owns a collection of Credential Sets for the different Reservation Systems they transact with.

External Resource

Any resources, such as tickets/vouchers, itineraries or other documents, that may be provided by the upstream Reservation System and/or Supplier with a Booking, which should be passed on to the customer.

Fare

The third and lowest level of differentiation of top-level Products into the ultimately bookable product variants (the other being BaseVariants and TimeSlots). Each Fare identifies a bookable pax or ticket type within a specific TimeSlot and thus BaseVariant of the top-level Product. While customers begin the checkout process by selecting a Product and departure or attendance date they wish to make a booking for, it is a selection of these Fares, in some cases combined with optional AddOns, that ultimately forms the concrete basis of the Booking.

Flow

A Flow is the step-by-step session bus containing all details and controlling the progression of the booking flow from selecting a top-level Product and desired travel date to a fully confirmed Booking.

From Price

The lowest undiscounted adult or general admission price observed across the future year or on a specific date (depending on context).
Useful as a lead-in or "window" price when listing product search results.

Invoice

Pricing and other details of the products purchased as part of a Booking. Essentially this is the result of the final Quote provided during checkout.

Line Item

A single line item in a Quote or Invoice breakdown.

Pax

The common industry jargon for passenger, also adopted for other areas of tourism to mean participant, guest, person etc.

Price

A container object for monetary amounts in combination with the applicable currency. Price and derived subtypes are used extensively throughout the API.

Price Preview

A displayable preview total price for a product selection, where possible complemented by a fully itemised Quote.

Product

Products are the many tours & activities which can be booked through the Livn API. Every Supplier may offer any number of products and the term refers to the top-level entities returned as the results of product discovery or search operations. Having chosen a specific product at the start of a new checkout flow, that product may spawn many variants, operate at different times or sessions and ultimately be available in a large number of bookable fare types, with distinct pricing, availability and conditions, which the customer will have to make a selection from.

Question

A request for information needed by the Reservation System from the end customer in order to proceed with the booking Flow.

Quote

A Quote is a detailed itemised breakdown of the customer's product selection with pricing information as well as terms of payment and cancellation. Used for both the price preview during the fare selection Step and the binding final quote that is issued by the Distributor and must be agreed to by the customer before committing to a booking with the upstream Reservation System.

Reservation System

One of the many online reservation systems used by the Suppliers found on the Livn API. Livn connects to these systems via their own individual API and it is part of the service to our clients that we maintain and upgrade our integration if and when a change on the upstream API necessitates development work.

Roadmap

The roadmap is an ordered list of Steps and Milestones you will find along the booking flow journey. The difference between Step and Milestone is that a Milestone is a Step that signifies a checkpoint along the journey. However, a Step itself is not necessarily a Milestone, since you might have to go through multiple steps until you reach a milestone.

Milestone is an enum with the values:

- FARE_SELECTION: The step in which the customer is asked to make a fareSelection of units to be purchased from the hierarchical list of available bookable product variants found in fareDetails.
- FINAL_QUOTE: Usually the last step before a confirmed booking or temporary reservation is written to the upstream reservation system. This Milestone includes the step.finalQuote, which includes as detailed as possible an itemised breakdown of the booking about to be made, including information on surcharges, transfer fees, local levies and the cancellation policy.
- TEMPORARY_HOLD: This is an optional milestone step that will not be present in every flow, but only where the supplier and their reservation system offer the facility to create a temporary reservation. This guarantees availability is held for a limited time without yet fully committing to the purchase.
- CONFIRMED_BOOKING: The end of every successful booking flow, at which point there are one or more flow.bookings records with associated booking.tickets.

Note: Apart from the TEMPORARY_HOLD Milestone, every checkout flow will include the other 3 milestones remaining (FARE_SELECTION,FINAL_QUOTE,CONFIRMED_BOOKING), and these will always be in that natural order. Between these Milestone steps, and theoretically, even before getting to the FARE_SELECTION milestone, there can be any number of other non-Milestone steps that depend entirely on the upstream reservation system or even a specific supplier or product. Such steps will always include a questions field, in fact, they only exist to ask these questions and collect additional customer input. These are questions that depend on the previous step, and thus cannot be asked in the previous step. At the same time, the user's input in response to these questions will decide what happens in the next step.

A good example is asking the customer for an optional or required hotel pick-up and drop-off. In some systems the list of available pick-up/drop-off locations may be fixed and not depend on the selected product variants, times and fares, so we can ask the customer to make a selection during the FARE_SELECTION milestone. In other systems, we may have to know for instance which time slot/session the customer wishes to book before we can ask the pick-up question. But in both cases that choice may impact on the final price, so we likewise cannot ask the pick-up question in the subsequent FINAL_QUOTE milestone. It has to be a separate step in between. Another existing example is a step needed in a specific upstream system, when booking optional add-ons, and not selecting a number of add-ons that matches the number of fares=pax in the group. In this case we need to ask the customer to specify which pax the add-ons are for, and this needs to happen after we collect the pax details, i.e. names etc, so the users can make that allocation.

While arguably a pick-up selection step might be something common enough that we could have made it a dedicated Milestone similar to the TEMPORARY_HOLD, that after all also doesn't always exist, we decided to leave these and more specialised steps with res-system-specific questions undefined.

In the roadmap, these non-milestone steps will appear in the following notation:

- [1] indicates that there is always exactly one additional step at that point of the flow
- [n] would indicate that there are always exactly *n* steps
- [1-2] would indicate at least one, but possibly two steps
- [0-*] indicates an unknown number of steps (including none at all) when we are simply unable to tell without knowing what input the customer makes in the previous milestone(s).

Select Option

A predefined selectable option that is an accepted Answer to a multiple-choice Question.

Seller

A Seller represents every client of Livn’s, whether they purely operate as a reseller buying from Livn Wholesale, purely as a Direct Connect client with a Channel into their own Distributor, or as a Hybrid of both. Every Seller can have multiple Channels to different Distributors, and Hybrid Sellers have a Direct Connect Channel into the Distributor that represents themselves plus a second Channel into the Livn Wholesale Distributor.

Step

A Step is the single logical step in the progression of a booking Flow, containing Questions, FareDetails, a final Quote and any other information that requires the customer's input to be passed on to the Supplier's Reservation System in order to proceed with the checkout. From the end user's perspective, each Step is the equivalent of a single screen in a booking application's user interface.

Supplier

Suppliers are companies that offer tours & activities products via one of the reservation systems Livn has connected to on the upstream. In most, though not all cases suppliers operate these products themselves, which is why we used to refer to them as Operators in v1 of the Livn API.

Ticket

A ticket that can be used to redeem all or part of a confirmed Booking.
Please Note: Some Suppliers may not accept third-party tickets generated using the details of Livn’s Ticket records, despite being part of the booking process with the Supplier's Reservation System. Booking records have an optional field for ExternalResources, which may include supplier-generated tickets, which are the only recognised admission document.

Time Slot

The second level of differentiation of top-level Products into the ultimately bookable product variants (the other being BaseVariants and Fares). Within the requested booking date, TimeSlots identify the different bookable departures or activity session start times and/or durations, within their parent ProductVariant, including a generic TimeSlot for open-timed products which do not make that distinction.

3. API Overview

The Livn API v2, like its predecessor, is built on industry-standard REST principles and uses JSON formatting for all request and response payloads. Null values are generally omitted from responses.

To start using the API you must first obtain authentication details by contacting Livn’s Tech Support via signup@livnapi.com.
Subsequent support is available once your organisation has entered into a commercial agreement with Livn over the use of our API.

3.1 Security

All communication between our clients and the Livn API endpoints is secured by SSL TLS 1.2+ encryption and all API requests require authentication and authorisation by either Basic HTTP authentication or JSON Web Tokens (JWT).

3.1.1 Basic HTTP authentication

All API clients will be set up in our system as a Seller and linked API User record and are initially issued a username and password that can only be used on the development and test endpoint https://dev.livnapi.com/public. Clients that have finalised all commercials and passed certification will receive a separate username and password for use on the production server or servers they have been assigned by Livn based on their requirements.

In either environment use the appropriate username and password pair to authenticate by including the HTTP request header of the form:

```
Authorization: Basic {Base64 encoded version of the concatenation {username}:{password}}
```

3.1.2 JSON Web Tokens (JWT)

For implementations that wish to make requests against the Livn API directly from a React, AngularJS or other browser-based JavaScript clients without exposing their username and password to the frontend client we offer an alternative authentication and authorisation pathway using JSON Web Tokens.

The process is to retrieve a token from the relative safety of a Node.js or similar client backend by sending the secret username and password to `POST /api/login`. The returned token which is valid for 3 hours but can be revoked by calling the `POST /api/logout` method at any time, can be used in the client by adding a different HTTP request header of the form:

```
Authorization: Bearer <token>
```

3.2 Product Search & Discovery

The products you will be able to access and book are based on the username and password used to authenticate the request, i.e. the API User record. Every API User is linked to a Seller account which, in turn, can be associated with one or more Channels. Each Channel links the Seller to a Distributor and gives the Seller and API User access to all or specific parts of that Distributor's inventory of Catalogues.

Products that are available in Catalogues owned by a Distributor that is effectively the same as the Seller, being connected by a Direct Connect Channel, will automatically be prioritised over identical products also offered by a third-party wholesaler, which generally means Livn Wholesale.

The development environment found on https://dev.livnapi.com/public can be, and unless required otherwise, will have been set up to perfectly mirror the production environment, meaning that the exact same Channels, Catalogues, Suppliers and Products should be available to API clients in both.

3.2.1 API search with product-specific departures

The most commonly used approach to begin every sales process is with a product search, making use of numerous filters based on Categories, geography (one or more `countries` and/or `latitude` & `longitude` or an IATA `airport` code plus a suitable search `radius`), duration, an operating date or date range or full-text search in products' names, descriptions or highlights.

In particular, for Products found as the result of a search that has not already been run to only include those known to operate on a specific date or date range, clients would next want to use the `GET /api/products/:productId/departures` call to retrieve the full calendar of the Product's Departures, i.e. the cached operating dates and lead-in FromPrices.

3.2.2 Search locally with cached inventory and operating dates

Some clients, in particular those that combine Livn's offering with other sources of products, prefer to keep a local copy of their accessible inventory and perform all search operations on their end.

Clients can use the calls on the `/api/suppliers` endpoints to fetch all or individual suppliers and `/api/products` to retrieve the list of Products of one specific supplier or individual Products available to them.

Alternatively, the Livn API provides a set of endpoints that can be used to populate and synchronise a local cache of Livn's inventory in bulk:

- The `GET /api/suppliers/productMinDetails` route, as the name suggests, returns only the minimal details of your accessible inventory in a single response: the list of supplier IDs at the root level and within each supplier the list of product IDs as well as a boolean flag to indicate if the product is temporarily disabled and cannot be booked.

```
[
    {
        "id": 2, <- Supplier ID
        "products": [
            {
                "id": 3, <- Product ID
                "disabled": false <- Product disabled status
            }
        ]
    },
    {
        "id": 3,
        "products": [
            {
                "id": 4,
                "disabled": true
            },
            {
                "id": 5,
                "disabled": true
            },
            {
                "id": 6,
                "disabled": false
            }
        ]
    },
    …
]
```

The `GET /api/products/operatingDates` endpoint returns the operating dates for all products in the user's accessible inventory as a single data dump using CSV formatting.If your implementation requires the full calendar of every product's operating dates, which would normally be used to perform date range or single date-specific client-side searches, we ask that you make use of this endpoint to populate and synchronize your local cache of all products' operating dates. The product-specific `GET /api/products/:productId/departures` call, which only returns data for a single product and which also includes the calculated date-specific FromPrices, should not be used to maintain a cache, but is intended for use at a time when a customer has already made the first decision to take a closer look at a specific product.

To reduce overhead, the call's response uses CSV formatting as it is produced directly from the underlying database, and unless explicitly requested otherwise we employ gzip compression. This is a small snippet of the response indicative of the formatting you can expect. Note that products, in particular those that are currently flagged as disabled, might not have any currently advertised departures / operating dates in the 366-day period Livn keeps track of.

```
productId,disabled,operatingDates
3,f,"{2023-03-05,2023-03-12,2023-03-19,2023-03-26,2023-04-02,2023-04-09,2023-04-16}"
4,t,
5,t,
6,f,"{2023-04-01,2023-04-02,2023-04-03,2023-04-04,2023-04-05,2023-04-06,2023-04-07,2023-04-08,2023-04-09}"
…
```

3.3 Wizard-style Checkout a.k.a. The Flow

The Livn API v2 Flow is the term used for both the sales session record and the underlying process which starts after the customer has chosen a product and departure/attendance date they wish to purchase, with the aim to complete a successful, fully confirmed Booking. As mentioned in the introduction above, the process is, in a nutshell, a wizard using a sequence of logical Steps to guide the API client and in turn the end customer through the checkout procedure in what is basically a back-and-forth of questions and answers exchanged between the API client, the Livn API and the Reservation System.

3.3.1 The Happy Path

To kick off the process, POST an empty Flow object to the API which initially only needs to contain the values `productId` and `date`. , and the optional currency to pick a specific booking currency where more than one is offered by the Supplier.

```
{
    "productId": 42,
    "date": "2021-05-24",
    "currency": "AUD"
}
```

Using these initial parameters together with your authentication details, the system will retrieve the best available instance of the specified Product from the Channels and Catalogues you have access to, automatically prioritising products found in the Direct Connect Channel. Unless the Product does not exist in the combined inventory accessible to the requesting user or has been temporarily disabled for the purpose of booking, the system now reaches out to the relevant Reservation System to collect all data on the available product variants and live pricing needed for the customer to make their product selection or otherwise determine which questions need to be answered prior to that in the first Step.

The Flow will be returned with a newly assigned id, a roadmap offering a guide to which Steps should be expected in the checkout process, as well as comprehensive details about the selected Product, its Supplier, the Catalogue, Channel, Distributor, and BillingInfo providing it, as well as the first Step containing instructions, fare details and/or questions, which now need to be presented to the customer.

This is an example of a typical booking Flow object in its first Step:

```
{
    "id": 394655,
    "livnReference": "99-394655",
    "productId": 1,
    "date": "2023-07-19",
    "currency": "AUD",
    "steps": [
        {
            "id": 472015,
            "status": "ACTIVE",
            "milestone": "FARE_SELECTION",
            "caption": "Select product fares.",
            "sequenceNumber": 1,
            "nextStepSequenceNumber": 2,
            "nextStepConfirmedBooking": false,
            "nextStepTemporaryHold": false,
            "allowBackHere": true,
            "stepName": "FARE_SELECTION",
            "created": "2022-07-20T23:09:57.990671Z",
            "modified": "2022-07-20T23:09:57.990672Z",
            "fareDetails": {
                "baseVariants": [
                    {
                        "timeSlots": [
                            {
                                "fares": [
                                    {
                                        "uuid": "00000001-0019-0007-0001-000000000000",
                                        "name": "STANDARD",
                                        "unitsAvailable": 22,
                                        "price": {
                                            "amount": 1349,
                                            "currency": "AUD"
                                        }
                                    }
                                ],
                                "availabilityShared": true,
                                "available": true
                            }
                        ],
                        "available": true
                    }
                ]
            }
        }
    ],
    "roadmap": "FARE_SELECTION,FINAL_QUOTE,CONFIRMED_BOOKING",
    "product": {
        "id": 1,
        "catalogueProductId": 5827,
        "disabled": false,
        "supplier": {
            "id": 1,
            "catalogueSupplierId": 844,
            "disabled": false,
            "name": "Intro Travel - Bali",
            "nameCompany": "Bali Intro",
            "businessNumber": "49 137 992 039",
            "description": "Intro Travel operate awesome group tours for 18-39 year olds in Asia & Australia. With a group leader from the moment you step off the plane, an instant group of friends and an amazing itinerary including the best of each destination. Get ready for the trip of a lifetime!",
            "website": "www.introtravel.com",
            "logo": {
                "url": "https://res.cloudinary.com/livngds/image/upload/v1621202238/V2/CENTRAL/supplierLogos/1/sup-1.png",
                "width": 200,
                "height": 150,
                "fileSize": 2848,
                "mimeType": "image/png",
                "title": "Intro Travel - Bali Logo"
            },
            "address1": "Level 14, 22 Market Street",
            "address2": "22 Market St",
            "city": "Sydney",
            "state": "New South Wales",
            "postcode": "2000",
            "country": "AU",
            "email": "sales@introtravel.com",
            "emailRes": "office@introtravel.com",
            "demo": false,
            "v1Cid": 664,
            "created": "2021-01-07T04:51:22.870183Z",
            "customMeta": null,
            "resSystem": "INTROTRAVEL"
        },
        "catalogue": {
            "id": 9,
            "name": "Test Catalogue for Certification"
        },
        "distributor": {
            "id": 2,
            "name": "Livn Holidays Wholesale Program",
            "company": {
                "id": 1,
                "code": "LIVN",
                "name": "Livn",
                "nameCompany": "Livn Pty Ltd",
                "businessNumber": "47 142 834 188",
                "address1": "49 Reservoir St",
                "city": "Surry Hills",
                "state": "NSW",
                "postcode": "2010",
                "country": "AU",
                "email": "info@livnapi.com",
                "phone": "+61292644411",
                "modified": "2021-01-11T05:14:04.008993Z"
            },
            "emailRes": "support@livnholidays.com",
            "created": "2021-01-11T05:17:06.531068Z",
            "customMeta": null
        },
        "channel": {
            "id": 2,
            "directConnect": true,
            "billingCurrency": "AUD",
            "created": "2021-01-18T05:27:47.19321Z"
        },
        "currencies": [
            "AUD",
            "EUR",
            "GBP",
            "USD"
        ],
        "name": "Bali Intro 9 Day Tour",
        "nameOriginal": "Bali Intro 9 Day Tour",
        "description": "If the time spent chilling on pure white sandy beaches, wandering through lush rice paddies and swimming with turtles sounds like your vibe, we think you’re going to fall hard for Bali Intro. The ultimate way to discover one of the world’s most tranquil and beautiful countries, expect everything from wild dolphin spotting to infinity pool chilling as you dive deep into Bali’s rich culture, learning to surf and discovering a love for delicious Nasi Goreng along the way. Oh, and did we mention you’ll be doing all this with a knowledgeable group leader and an instant crew of new mates? The trip of a lifetime just got even better.",
        "itinerary": {
            "items": [
                {
```

```json
                    "dayFrom": 1,
                    "dayTo": 1,
                    "title": "Day 1: Arrive in Canggu",
                    "body": "Landing in Denpasar, we’ll meet you at the airport and take you to your hotel in chilling Canggu. Meet up with your travel crew and get excited for the adventures to come."
                },
                {
                    "dayFrom": 2,
                    "dayTo": 2,
                    "title": "Day 2: Temples & Beach Sunsets",
                    "body": "Today’s all about ancient temple exploring and sunsets on the beach as you get the first taste of Bali’s beautiful culture. Tonight, we hit the dancefloor."
                },
                {
                    "dayFrom": 3,
                    "dayTo": 3,
                    "title": "Day 3: Learn to Surf",
                    "body": "Canggu is known as one of the world’s most iconic surf destinations, and you’ll soon see why as you hit the waves with the help of local Balinese surf instructors."
                },
                {
                    "dayFrom": 4,
                    "dayTo": 4,
                    "title": "Day 4: Exploring Lovina",
                    "body": "Untouched and relatively unknown, Lovina will steal your heart with its rolling rice paddies and smiley welcome. On the way, we’ll squeeze in some temple exploring and a dip in a waterfall before arriving at our unique accommodation set high in the mountains with views of rice paddies and the shimmering ocean."
                },
                {
                    "dayFrom": 5,
                    "dayTo": 5,
                    "title": "Day 5: Dolphin Spotting & Local Life",
                    "body": "After brekkie on the beach, it’s all aboard a traditional outrigger boat as you cruise the coastline on the lookout for wild dolphins. Back on land, we’ll then visit a local family where you’ll learn the art of basket weaving and get a unique insight into traditional Balinese life. Keen to chill? The infinity pool of our luxury villa has your name all over it."
                },
                {
                    "dayFrom": 6,
                    "dayTo": 6,
                    "title": "Day 6: Hello Paradise",
                    "body": "Crystal clear waters and swaying palm trees are on the agenda today as we catch the fast ferry to Gili T, the largest of the three Gili Islands. With a free afternoon feel yourself slipping easily into island time, Bintang in hand, before a night out island style."
                },
                {
                    "dayFrom": 7,
                    "dayTo": 7,
                    "title": "Day 7: Time to Chill",
                    "body": "With no motorised vehicles on Gili T, today you’ll explore the island on two wheels, beach hopping from one sandy shore to the next before capturing a sunset snap-on THAT iconic ocean swing."
                },
                {
                    "dayFrom": 8,
                    "dayTo": 8,
                    "title": "Day 8: Snorkelling Adventure",
                    "body": "Travelling by boat, today you’ll explore the beauty of all three Gili Islands, stopping along the way for plenty of snorkel breaks to spot turtles and other colourful marine life."
                },
                {
                    "dayFrom": 9,
                    "dayTo": 9,
                    "title": "Day 9: Depart Bali or Extend Your Stay",
                    "body": "Your unforgettable 9-day adventure comes to an end today. Those flying home will make their way to the airport, or keep living the dream and extend your stay in Bali with our help and the new friends you’ve made."
                }
            ]
        },
        "inclusions": {
            "items": [
                {
                    "type": "ACCOMMODATION",
                    "content": "8 nights accommodation"
                },
                {
                    "type": "MEALS",
                    "content": "7 Breakfasts, 1 Lunch, 2 Dinners"
                },
                {
                    "type": "TRANSPORT",
                    "content": "Personal Airport pickup to the hotel 24/7"
                },
                {
                    "type": "TRANSPORT",
                    "content": "All Transport"
                },
                {
                    "type": "OTHER",
                    "content": "Group Leader"
                },
                {
```

```
            "content": "Canang Sari (Local Offering) Making Class"
        },
        {
            "content": "Bahasa Language Lesson"
        },
        {
            "content": "Soak up Stunning Sunsets"
        },
        {
            "content": "Experience Bali Nightlife"
        },
        {
            "content": "Learn to Surf and Swim under Waterfalls"
        },
        {
            "content": "Visit a local Balinese Family"
        },
        {
            "content": "Basket Weaving"
        },
        {
            "content": "Coffee Plantation & Rice Paddies"
        },
        {
            "content": "Dolphin Boat Trip"
        },
        {
            "content": "Explore Tropical Islands by Bicycle"
        },
        {
```

```
                    "content": "Boat Trip around all 3 Gili Islands"
                },
                {
                    "content": "Snorkel with Turtles"
                },
                {
                    "content": "Sunset Swing & Island Party"
                },
                {
                    "content": "Help with Onward Travel"
                }
            ]
        },
        "highlights": {
            "highlights": [
                "Learn to surf with Balinese surfer dudes before kicking back with a sunset beer",
                "Hop on board a traditional outrigger boat and spot pods of wild dolphins",
                "Snorkel with turtles and hang out in paradise on all 3 Gili Islands"
            ]
        },
        "pickupNotes": "We will pick you up from the airport and take you to the accommodation in Canggu.",
        "dropoffNotes": "The tour ends in Gili Trawangan.",
        "specialNotes": "Conditions:\nThis tour is recommended for 18 to 39 years. It’s best to arrive on the actual start date.
It’s also recommended to have onward travel booked out of Bali before travelling there, as you may get asked for proof of this
before boarding your flight to Bali. Most nationalities including the UK, Europe, Australia, NZ, Canada, US, Scandinavia and Brazil
will usually receive a 30-day visa on arrival in Bali. This visa will be free for most nationalities and at a cost of approx. $35USD
for others. We recommend checking with your local Indonesian consulate for up to date visa requirements and costs. If you need any
further help with visas just contact us. You must be at least 18 years old, or if you are under 18 years old, obtain the consent of,
or during the Tour be under the supervision of, a legal guardian that is at least 18 years old. Participation in our Tours may
involve physical effort and exertion. You must not participate in a Tour if participation in the Tour will adversely affect your
```

```
physical health and/or medical condition. We recommend that you seek medical advice regarding your participation in any Tours. Your
participation in a Tour will be at your own risk, and you agree to sign a waiver form prior to your participation in any Tours. You
are responsible for obtaining any vaccinations or medical information relevant to your destination country. You must have a valid
passport for the duration of any Tour and for 6 months thereafter. It is your responsibility to have a valid visa throughout the
duration of the Tour. We will not be held responsible if you are denied entry to a destination country due to an invalid passport or
visa.\n\nInsurance:\nTravel Insurance for Bali is compulsory to join Bali Intro. Your group leader will ask to see proof of your
cover before your trip begins.\n\nNot included:\nFlights, travel insurance, visas and some meals.\n\nCOVID-19, Health &
Safety:\nYour destination or airline may have COVID-19 vaccination or testing requirements to enter. It is your responsibility to
adhere to these requirements to ensure your arrival is not impacted and you can join your tour without issue. There are no
compulsory vaccinations for any of our destinations and all areas we travel to are very low risk. But if you are unsure or need
advice we recommend visiting your local GP.",
        "groupSizeMax": 22,
        "duration": 12960,
        "ageMin": 18,
        "ageMax": 39,
        "fromPrices": [
            {
                "amount": 1349,
                "currency": "AUD"
            },
            {
                "amount": 899,
                "currency": "EUR"
            },
            {
                "amount": 749,
                "currency": "GBP"
            },
            {
                "amount": 999,
                "currency": "USD"
            },
            {
                "amount": 1449,
                "currency": "NZD"
            }
        ],
        "operatingDays": 127,
        "operatingDaysStr": "Mon,Tue,Wed,Thu,Fri,Sat,Sun",
        "locationsStart": [
            {
                "latitude": -8.74494158235058,
                "longitude": 115.16706031089188,
                "tz": "Asia/Makassar",
                "continent": "AS",
                "city": "Badung",
                "district": "Kuta",
                "country": "ID",
                "streetAddressAccuracy": true,
                "business": "Ngurah Rai International Airport",
                "address1": "Jalan Raya Gusti Ngurah Rai, Tuban",
                "postcode": "80361"
            }
        ],
        "locationsEnd": [
            {
                "latitude": -8.351844,
                "longitude": 116.0561632,
                "tz": "Asia/Makassar",
                "continent": "AS",
                "city": "Lombok Utara",
                "district": "Pemenang",
                "country": "ID",
                "streetAddressAccuracy": false,
                "postcode": "83756"
            }
        ],
        "images": [
            {
                "url":
"https://res.cloudinary.com/livngds/image/upload/v1610507400/V2/CENTRAL/productImages/1/l8wvmlou3yutgikiez7a.jpg",
                "width": 800,
                "height": 600,
                "fileSize": 53524,
                "mimeType": "image/jpg"
            },
            {
                "url":
"https://res.cloudinary.com/livngds/image/upload/v1645741252/V2/CENTRAL/productImages/1/ftgubdwvwhxjxutnltve.jpg",
                "width": 800,
                "height": 600,
                "fileSize": 42597,
                "mimeType": "image/jpg"
            },
            {
                "url":
"https://res.cloudinary.com/livngds/image/upload/v1645741253/V2/CENTRAL/productImages/1/k1frtbrceq9mxoid8imx.jpg",
                "width": 1069,
                "height": 802,
                "fileSize": 107976,
                "mimeType": "image/jpg"
            },
            {
                "url":
"https://res.cloudinary.com/livngds/image/upload/v1645741253/V2/CENTRAL/productImages/1/pff7ig46st2foqcmdjwz.jpg",
                "width": 1069,
                "height": 802,
                "fileSize": 102549,
                "mimeType": "image/jpg"
            },
            {
                "url":
"https://res.cloudinary.com/livngds/image/upload/v1645741253/V2/CENTRAL/productImages/1/wavaqiqtsgzxvilf7qep.jpg",
                "width": 1069,
                "height": 802,
                "fileSize": 74975,
                "mimeType": "image/jpg"
            }
        ],
        "categories": [
            {
                "id": 14,
                "name": "Multi Day & Extended Touring"
            },
            {
                "id": 15,
                "name": "National Parks & Natural Attractions"
            },
            {
                "id": 7,
                "name": "Cruises, Sailing & Water Sports"
            }
        ],
        "demo": false,
        "v1Cid": 38479,
        "created": "2021-01-07T04:54:15.237139Z",
        "commissionPerc": 0,
        "resSystem": "INTROTRAVEL",
        "usesNetRates": false,
        "customMeta": null
    },
    "simulation": true,
    "directConnect": true,
    "billingInfo": {
        "billingCurrency": "AUD",
        "billingExchangeRate": 1
    },
    "customMeta": null,
    "answers": {},
    "errors": [],
    "step": {...}
}
```

At any point of the checkout process, in order to progress the Flow, the customer through your API client needs to supply the necessary information asked for in the Flow’s Step currently marked as `ACTIVE`.

These are the elements the client needs to look for in a Step to determine what customer input is needed to fulfil the “question and answer” mechanism of our flow (see corresponding sample JSON in the table below):

- FareDetails → FareSelection/AddOnSelection:

  If the Step contains the `fareDetails` field clients must provide the customer’s Fare and optionally AddOn selection by including the `fareSelections` and `addOnSelections` in the same Step when POST-ing back the flow object.

```
{
    "id": 472287,
    "status": "DONE",
    "milestone": "FARE_SELECTION",
    "caption": "Select product fares.",
    "sequenceNumber": 1,
    "nextStepSequenceNumber": 2,
    "nextStepConfirmedBooking": false,
    "nextStepTemporaryHold": false,
    "allowBackHere": true,
    "stepName": "FARE_SELECTION",
    "created": "2022-07-21T03:01:22.420575Z",
    "modified": "2022-07-21T03:01:30.532235Z",
    "fareDetails": {
        "baseVariants": [
            {
                "timeSlots": [
                    {
                        "fares": [
                            {
                                "uuid": "00000001-0019-0007-0001-000000000000",
                                "name": "STANDARD",
                                "unitsAvailable": 22,
                                "price": {
                                    "amount": 1349,
                                    "currency": "AUD"
                                }
                            }
                        ],
                        "availabilityShared": true,
                        "available": true
                    }
                ],
                "available": true
            }
        ]
    },
    "fareSelections": [
        {
            "uuid": "00000001-0019-0007-0001-000000000000",
            "hint": "null > null > STANDARD",
            "quantity": 2
        }
    ]
}
```

- Questions → Answers

For any Step containing a `questions` field clients must collect and provide the customer's input by including the `answers` in the same Step when POST-ing back the flow object.

```
{
    "id": 472288,
    "status": "DONE",
    "milestone": "FINAL_QUOTE",
    "caption": "Submit passenger details, accept final quote, and confirm your booking.",
    "sequenceNumber": 2,
    "previousStepSequenceNumber": 1,
    "nextStepSequenceNumber": 3,
    "nextStepConfirmedBooking": true,
    "nextStepTemporaryHold": false,
    "allowBackHere": true,
    "previousStep": "FARE_SELECTION",
    "stepName": "FINAL_QUOTE",
    "expires": "2022-07-21T03:16:30.470045Z",
    "created": "2022-07-21T03:01:30.521054Z",
    "modified": "2022-07-21T03:02:17.178037Z",
    "questions": {
        "questionGroups": [
            {
                "caption": "Passenger Details - STANDARD - Passenger 1",
                "questions": [
                    {
                        "uuid": "00000001-0019-0007-0002-000000000000",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "PAX_FIRST_NAME",
                        "title": "First Name",
                        "example": "John",
            "required": true
        },
        {
            "uuid": "00000001-0019-0007-0002-000000000001",
            "fareUuids": [
                "00000001-0019-0007-0001-000000000000"
            ],
            "scope": "PER_GROUP",
            "answerType": "TEXT",
            "purpose": "PAX_LAST_NAME",
            "title": "Last Name",
            "example": "Smith",
            "required": true
        },
        {
            "uuid": "00000001-0019-0007-0002-000000000002",
            "fareUuids": [
                "00000001-0019-0007-0001-000000000000"
            ],
            "scope": "PER_GROUP",
            "answerType": "EMAIL",
            "purpose": "PAX_EMAIL",
            "title": "E-mail",
            "example": "john.smith@email.com",
            "required": true
        },
        {
            "uuid": "00000001-0019-0007-0002-000000000003",
            "fareUuids": [
                "00000001-0019-0007-0001-000000000000"
```

```
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "PHONE",
                        "purpose": "PAX_PHONE",
                        "title": "Phone Number",
                        "description": "Phone number",
                        "example": "+61491570156",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000004",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "DATE",
                        "purpose": "PAX_DOB",
                        "title": "Date of Birth",
                        "description": "Please answer in yyyy-MM-dd format.",
                        "example": "1975-01-25",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000005",
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "SUPPLIER_NOTES",
                        "title": "Comments",
                        "description": "General booking notes or comments you wish to pass to the supplier",
                        "example": "I'm allergic to shellfish",
                        "required": false
                    }
                ]
            },
            {
                "caption": "Passenger Details - STANDARD - Passenger 2",
                "questions": [
                    {
                        "uuid": "00000001-0019-0007-0002-000000000006",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "PAX_FIRST_NAME",
                        "title": "First Name",
                        "example": "John",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000007",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "PAX_LAST_NAME",
                        "title": "Last Name",
                        "example": "Smith",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000008",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "EMAIL",
                        "purpose": "PAX_EMAIL",
                        "title": "E-mail",
                        "example": "john.smith@email.com",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000009",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "PHONE",
                        "purpose": "PAX_PHONE",
                        "title": "Phone Number",
                        "description": "Phone number",
                        "example": "+61491570156",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000010",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "DATE",
                        "purpose": "PAX_DOB",
                        "title": "Date of Birth",
                        "description": "Please answer in yyyy-MM-dd format.",
                        "example": "1975-01-25",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000011",
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "SUPPLIER_NOTES",
                        "title": "Comments",
                        "description": "General booking notes or comments you wish to pass to the supplier",
                        "example": "I'm allergic to shellfish",
                        "required": false
                    }
                ]
            }
        ]
    },
    "answers": {
        "answers": [
            {
                "questionUuid": "00000001-0019-0007-0002-000000000000",
                "value": "John"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000001",
                "value": "Smith"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000002",
                "value": "john.smith@email.com"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000003",
                "value": "+61491570156"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000004",
                "value": "1975-01-25"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000005",
                "value": "I'm allergic to shellfish"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000006",
                "value": "Jane"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000007",
                "value": "Smith"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000008",
                "value": "jane.smith@email.com"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000009",
                "value": "+61491570157"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000010",
                "value": "1976-10-20"
            }
        ]
    },
    "finalQuote": {...
}
```

All types of user inputs i.e. answers as well as fare and add-on selections, use the `uuid` property of their respective counterparts to reference a question, fare or add-on. During the fare selection step, a question may also be associated with specific fares and this could be determined via the `question.fareUuids` property.

Steps containing neither `fareDetails` nor `questions` do not require any manual input from the customer and merely serve as a stopping point with the checkout flow being in a specific state, e.g. having secured a time-limited temporary reservation before confirming the booking. For these steps, it is sufficient to simply POST back the Flow object without any additions. We are generally trying to minimise the number of Steps needed to complete the checkout, by asking for user input at the best suitable time..

Once all necessary inputs have been supplied and POST-ed back to Livn, the system will validate the flow's current step. This also includes a check that the current step's `expiry` timestamp (if it exists) has not been exceeded. If all validations pass we will again communicate with the upstream reservation system, to work out the requirements for the next step. In response, the client will receive the updated flow with the next `ACTIVE` step to act on.

We would repeat this process until the response you get is a flow object containing all steps with `status:DONE` and the last step with `milestone:CONFIRMED_BOOKING`. This signifies that you have completed the booking flow. Once you reach this point, the flow will contain the booking details including tickets and an invoice.

3.3.2 The fingerprintHash and dnaHash

In May 2023 we introduced a pair of new properties to a set of types that make up the `Step.fareDetails` and `Step.questions` models used during checkout:

- `BaseVariant`, `TimeSlot`, `Fare` and `AddOn`
- `QuestionGroup`, `Question` and `SelectOption`

The new properties added to all of the above types are called `fingerprintHash` and `dnaHash` and take the form of SHA-256 hash strings.

A `fingerprintHash` is a hash of the combination of primary identifiers used by the supplier's reservation system that distinguishes a record, such as a `BaseVariant`, `Fare` or `Question`, **from its siblings**. Related records at the same level but with different parents, e.g. the *Adult* `Fare` types sitting under various different `BaseVariant-TimeSlot` branches, will commonly have the same `fingerprintHash` but each will have a distinct `dnaHash`.

A `dnaHash` is the unique hash of a record's fingerprint, so the same combined properties used to identify it in the supplier's subsystem forming the basis of its `fingerprintHash`, but in conjunction with the fingerprints of its entire line of parents/ancestors. Effectively, this results in a signature that identifies a single specific node in an entire `Step.fareDetails` or `Step.questions` tree that is unique within the entire scope of the relevant top-level `Product`.

We are excited to release this new feature as it is a prerequisite for a host of useful applications that require the ability to identify, and thus recognise at a later time, instances of the above model types. Unlike `Products`, which are imported, edited and enriched by Livn in a process we call Content Curation, the `BaseVariant`, `TimeSlot`, `Fare` etc objects appearing during checkout are all generated on-the-fly from data received directly from the relevant upstream reservation system.

The existing `uuid` primary identifiers, which are used to refer to some of these types in the scope of the Livn API, namely to define `Step.fareSelections`, `Step.addOnSelections` and `Step.answers`, were never intended to be used as lasting identifiers beyond the scope or lifetime of the `Flow` they appear in. These `uuid` strings are either true, randomly generated UUIDs or are sequentially generated "pseudo-UUIDs" that merely borrow the 8-4-4-4-12 character UUID format. Either way, there is no guarantee that a particular `Fare` or `Question` with a specific `uuid` value from one `Flow` will appear with the same `uuid` in another `Flow`, be it for the same `Flow.date` or another date.

Prior to the addition of these hashes, clients of our API who wish to keep track of the available variants, sessions and fare types of products as they appear over time, had to resort to using the various human-readable properties like `name`, `description`, `title` etc to try and identify these by. However, just like the `uuid` properties, which also don't exist at all levels of the `FareDetails` and `Questions` model, these text fields are ill-suited to serve as lasting identifiers. These fields are prone to change, be it due to code changes made by Livn to fix or improve the way data coming from a supplier is presented downstream or simply by the supplier making a change on their end. The new `fingerprintHash` and `dnaHash` values, on the other hand, are based on values that are supposed to be static for the lifetime of these records in their originating system, so ideally values like database primary keys published by the reservation system API and used there to refer to a product session, fare type or question. The expectation is that a record with the same fingerprint and DNA (and thus the derived hashes) represents the same entity in the connected reservation system while a changed `fingerprintHash` and `dnaHash` indicate a different or new entity upstream. The exact combination of properties or the nature of these fingerprints is specific to the different reservation systems we have connected, but the hashes we produce for each all have the same format.

If your implementation depends on the ability to identify and keep track of data objects from our checkout-time API model, you'll want to work with the `fingerprintHash` and `dnaHash` properties. We already have a few upcoming features for our API in the development pipeline that weren't feasible, or technically possible, without this important step of adding reliable fingerprinting at every relevant level of the checkout-time data model. The first such extension we have planned requires the new ability to track sub-`Product`-level information about product variants, time slots and fares in a perpetual cache record and we hope to be sharing this data with our clients soon.

Finally, here is a code example of a plausible `Step.fareDetails` tree with reduced details that illustrates the difference and relationship between `fingerprintHash` and `dnaHash` values.

```
{
    "baseVariants": [
        {
            "name": "Standard Hot Air Balloon Tour",
            "fingerprintHash": "ab63ab7b7352d624f8c278bcb518d06c8e273b05e58ee9393fec90654c7f13f6", // B
            "dnaHash": "95a58d9acf089730e628b0b7e56388fe67693386d9fdbbcc93d0b1e07f04a386", // B.P
            "timeSlots": [
                {
                    "name": "06:00",
                    "fingerprintHash": "e5981171ff05f70b821f36acb5bc7fa155d43186bd8b434094ce6eff1bb6cf5f", // T
                    "dnaHash": "45f0493f2477895e24a5dcffdb36e4237ae086a61c6f3df7dc2af1f1c9bbc900", // T.B.P
                    "fares": [
                        {
                            "uuid": "5bd42159-c164-4f17-b72d-3027560e4bf3",
                            "name": "Adult",
                            "price": {
                                "amount": 279,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "1e4cfe06205a7eb4cccbc0e58122e4f686f3b7c756f1a0c420cc3ba86fac615c", // F
                            "dnaHash": "626b2968be23c6c54703df642c79a6409975302eff7fb8e1f57004ec23fc66fe" // F.T.B.P
                        },
                        {
                            "uuid": "3ff656ee-51d4-47d8-ab5c-6a83dcd1d0e1",
                            "name": "Child",
                            "price": {
                                "amount": 239,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "b8febd3bf366b40a9b2065a6247426d5feacfffdd049efdadaba2e1e370b5778", // F
                            "dnaHash": "0b80b492a625c51ebc914404a9bb63fee72fe5160652092b2d7318754c7c186d" // F.T.B.P
                        },
                        {
                            "uuid": "c86adc74-4196-4743-8133-6463de009b6e",
                            "name": "Family (2 Adults + 2 Children)",
                            "price": {
                                "amount": 929,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "bdd8fd991cb5c58a73dceef8f1acebc52f996984b1d8510bc49c6ce0ab8decd5", // F
                            "dnaHash": "d6e5fec1f0e2cdc93434eeff982061b8205737d7ef188a916e4b73eeb28d0d5d" // F.T.B.P
                        },
                        {
                            "uuid": "689c3afd-c1ed-4176-9a9d-93053dddc9a6",
                            "name": "Senior",
                            "price": {
                                "amount": 259,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "59544d7b4a80ba727390c1dddf32689e498205b0af04f63d5ce9a62b11b0a015", // F
                            "dnaHash": "d4c87b638c3162082b3c5584d48d9ec89d049f9592e9001038a21225fbacbd3e" // F.T.B.P
                        }
                    ]
                },
                {
                    "name": "10:00",
                    "fingerprintHash": "0ae241d4188e1f01801700f6eaa8e16efcff83f5ebbe8c57a5f38919ab32c285", // T
                    "dnaHash": "dfe4d8a2c6a2749935244c2ebf4bdbd32d2eb8cc76704c8f4200ae5eebdb4ff4", // T.B.P
                    "fares": [
                        {
                            "uuid": "cd48d73c-536f-4f43-a94f-7cc13e1e50de",
                            "name": "Adult",
                            "price": {
                                "amount": 259,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "1e4cfe06205a7eb4cccbc0e58122e4f686f3b7c756f1a0c420cc3ba86fac615c", // F
                            "dnaHash": "7118e648b4821411c8e0a04c1b26eac5a2b3b1a58cc7df6e7a9f092472d31e04" // F.T.B.P
                        },
                        {
                            "uuid": "05e708f4-da73-41cb-ba2d-66bf757dafac",
                            "name": "Child",
                            "price": {
                                "amount": 219,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "b8febd3bf366b40a9b2065a6247426d5feacfffdd049efdadaba2e1e370b5778", // F
                            "dnaHash": "4df95bcadfe4b8561b729ee3adba6b14b67751e883457929256312e6bf555f24" // F.T.B.P
                        },
                        {
                            "uuid": "0f865553-23ec-4867-b629-c954d8c62ed3",
                            "name": "Family (2 Adults + 2 Children)",
                            "price": {
                                "amount": 849,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "bdd8fd991cb5c58a73dceef8f1acebc52f996984b1d8510bc49c6ce0ab8decd5", // F
                            "dnaHash": "6cc248922a10edaab114d2104a322ef3b8858087dcc321aa1026655d7ea57ed0" // F.T.B.P
                        },
                        {
                            "uuid": "ef0edb90-a42b-4678-b3db-476458c4e2f3",
                            "name": "Senior",
                            "price": {
                                "amount": 239,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "59544d7b4a80ba727390c1dddf32689e498205b0af04f63d5ce9a62b11b0a015", // F
```

```
                            "dnaHash": "9c6bc4975ced0f74aab3f3bce65ded1c4e024ac18309312fb26292c4a56edcd2" // F.T.B.P
                        }
                    ]
                }
            ]
        },
        {
            "name": "Deluxe Hot Air Balloon Tour with Breakfast in the Sky",
            "fingerprintHash": "ef48133c679c02dd1a4b2c0e968ea5e3c2f5e4c79a9f51d03791664d4de7f9d0", // B
            "dnaHash": "bd929558449807fe6676ccda9f14606c8a5ab9bacc4f964c439730cc049cf3bc", // B.P
            "timeSlots": [
                {
                    "name": "06:00",
                    "fingerprintHash": "e5981171ff05f70b821f36acb5bc7fa155d43186bd8b434094ce6eff1bb6cf5f", // T
                    "dnaHash": "a3eec9ab6dbd92d3fabad3ac0a7eeb5e0ed152ce3918be5e5a97b88951e9a5d1", // T.B.P
                    "fares": [
                        {
                            "uuid": "de248f3a-a59d-4537-8b47-d981a1cbc326",
                            "name": "Adult",
                            "price": {
                                "amount": 309,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "1e4cfe06205a7eb4cccbc0e58122e4f686f3b7c756f1a0c420cc3ba86fac615c", // F
                            "dnaHash": "1619fc4bda3be1957f98d8115c5a5e84fb1cdfe4247925cc306eb9e611c6ebc3" // F.T.B.P
                        },
                        {
                            "uuid": "7531820e-28be-497e-a476-0e43e991b7ce",
                            "name": "Child",
                            "price": {
```

```
                "amount": 269,
                "currency": "AUD"
            },
            "fingerprintHash": "b8febd3bf366b40a9b2065a6247426d5feacfffdd049efdadaba2e1e370b5778", // F
            "dnaHash": "09f8814acd14ce77a5f21af3cd26f7c9db7224a187a16a26f3206ffb2007f891" // F.T.B.P
        },
        {
            "uuid": "127b3447-63f1-47b6-9dc6-e78e5c4632d3",
            "name": "Family (2 Adults + 2 Children)",
            "price": {
                "amount": 999,
                "currency": "AUD"
            },
            "fingerprintHash": "bdd8fd991cb5c58a73dceef8f1acebc52f996984b1d8510bc49c6ce0ab8decd5", // F
            "dnaHash": "822bae67d7d2dd1e65c879b51facf22ca8d7cb41db5bd1159eac362406f73d3c" // F.T.B.P
        },
        {
            "uuid": "0be5098c-6c28-4d06-9152-864730e3a652",
            "name": "Senior",
            "price": {
                "amount": 289,
                "currency": "AUD"
            },
            "fingerprintHash": "59544d7b4a80ba727390c1dddf32689e498205b0af04f63d5ce9a62b11b0a015", // F
            "dnaHash": "817a65c42033c7fce887bde8739e63097642ed8f30180f4e09804f0d4e8337de" // F.T.B.P
        }
    ]
},
{
    "name": "10:00",
    "fingerprintHash": "0ae241d4188e1f01801700f6eaa8e16efcff83f5ebbe8c57a5f38919ab32c285", // T
    "dnaHash": "4179899cfd444268adc898c15667856007a6a5ddf6b17ad24816c28efb8b5bb8", // T.B.P
    "fares": [
        {
            "uuid": "88ac53dc-2a63-4685-90b1-aaf087487fb4",
            "name": "Adult",
            "price": {
                "amount": 259,
                "currency": "AUD"
            },
            "fingerprintHash": "1e4cfe06205a7eb4cccbc0e58122e4f686f3b7c756f1a0c420cc3ba86fac615c", // F
            "dnaHash": "ded5c85a8afabd4d300e3391208e9d3cec98a9d47b8558cc22f50165cd8249cd" // F.T.B.P
        },
        {
            "uuid": "c338f0d8-f11c-4854-a282-a97d1c678097",
            "name": "Child",
            "price": {
                "amount": 219,
                "currency": "AUD"
            },
            "fingerprintHash": "b8febd3bf366b40a9b2065a6247426d5feacfffdd049efdadaba2e1e370b5778", // F
            "dnaHash": "3c2fc0e1807eb3b035f313af241348846f580737c0b1f6446f0badb4a7a4aa95" // F.T.B.P
        },
        {
            "uuid": "9f3fcbd2-5bde-49fc-a93e-a773a55a834a",
            "name": "Family (2 Adults + 2 Children)",
            "price": {
                "amount": 849,
                "currency": "AUD"
            },
                            "fingerprintHash": "bdd8fd991cb5c58a73dceef8f1acebc52f996984b1d8510bc49c6ce0ab8decd5", // F
                            "dnaHash": "441037bb6268badc0a50fbb9a6818bfaac14e3154620d8e704a64dd574a2518d" // F.T.B.P
                        },
                        {
                            "uuid": "4970410b-8cce-4259-9a15-dd0ad11f5c54",
                            "name": "Senior",
                            "price": {
                                "amount": 239,
                                "currency": "AUD"
                            },
                            "fingerprintHash": "59544d7b4a80ba727390c1dddf32689e498205b0af04f63d5ce9a62b11b0a015", // F
                            "dnaHash": "b96aa68a03e881bd8c19ade6c91fe5c8639ebd095f872a72ab0367b290a33a4a" // F.T.B.P
                        }
                    ]
                }
            ]
        }
    ]
}
```

As you can see, the same four basic fare types appear four times under the two base variants and each of their two time slots. Each instance of the same fare has the same `fingerprintHash`, e.g. *Adult* fares all have the same:

```
 "fingerprintHash": "1e4cfe06205a7eb4cccbc0e58122e4f686f3b7c756f1a0c420cc3ba86fac615c", // F
```

…but their `dnaHash` is different in each case, because this includes the difference in fingerprints of their respective parent time slots and, in turn, their respective parent base variant, e.g.:

```
"dnaHash": "ded5c85a8afabd4d300e3391208e9d3cec98a9d47b8558cc22f50165cd8249cd" // F.T.B.P
```

vs

```
"dnaHash": "626b2968be23c6c54703df642c79a6409975302eff7fb8e1f57004ec23fc66fe" // F.T.B.P
```

…the former uniquely identifies the *Adult* `Fare` under the *10:00* `TimeSlot` of the *Deluxe* `BaseVariant` of the selected top-level `Product`, while the latter leads to the *Adult* `Fare` under the *06:00* `TimeSlot` of the *Standard* `BaseVariant`.

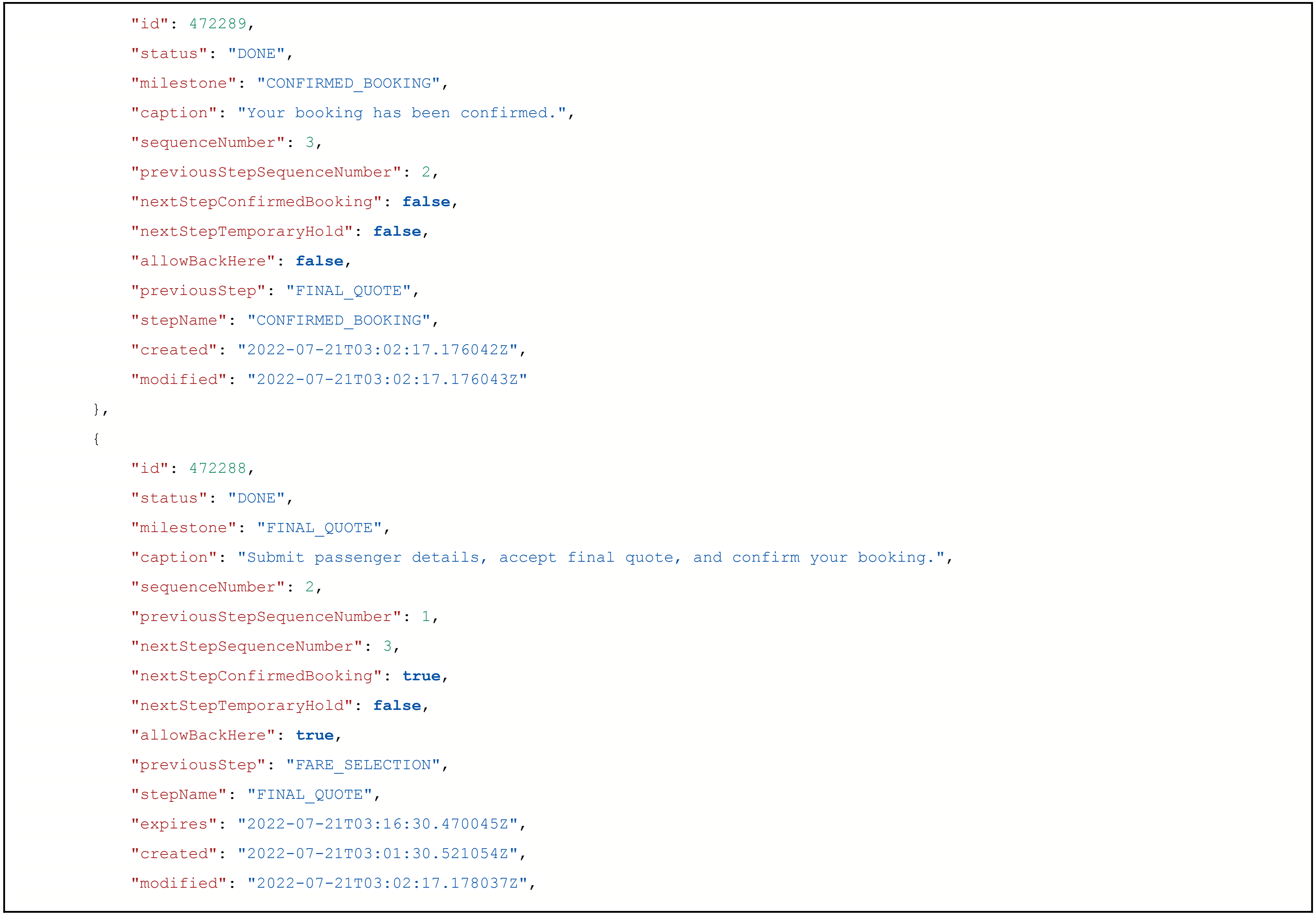
So the two hashes each have their own unique use cases or applications. The `fingerprintHash` values can be used to draw connections between related instances of the same basic types. As with the *Adult* fares above, any two instances of the same type that share a `fingerprintHash` are referred to using the same primary identifiers in the system used by the product supplier making them related. Meanwhile, `dnaHash` values serve a different purpose in that these allow identifying a singular specific object within the scope of the `Product` that is being booked. A possible use case would be to replay a concrete fare selection made by a customer at some point, but in an entirely new checkout process and `Flow`, incuding on a different travel date. Needless to say, not every `BaseVariant`, `TimeSlot` and `Fare` will necessarily be available on every operating date of the `Product`. Likewise, fares can sell obviously sell out or otherwise become unavailable over time. So there is no guarantee that a previous fare selection is still valid and can be booked for a different date or even the same date but at a later time, nor that the prices quoted originally haven't changed.

3.3.3 Moving backwards "against the flow"

In the event a customer wishes to go back to a previous step to change some selections or answers, you can do this by using the request parameter `back` and POSTing the unchanged flow object back to the API (`POST /api/flows?back=true`).

Please note that depending on the way the checkout process works on the upstream reservation system, this request will not in all cases move back to the immediately previous step. The step class has variables called `sequenceNumber`, an incremental integer numbering all the steps you have encountered in the flow, and `previousStepSequenceNumber`, to indicate the step you can move back to and make changes.

```
{
    "id": 394908,
    "livnReference": "99-394908",
    "productId": 1,
    "date": "2023-07-19",
    "currency": "AUD",
    "steps": [
        {
```



```
            "id": 472289,
            "status": "DONE",
            "milestone": "CONFIRMED_BOOKING",
            "caption": "Your booking has been confirmed.",
            "sequenceNumber": 3,
            "previousStepSequenceNumber": 2,
            "nextStepConfirmedBooking": false,
            "nextStepTemporaryHold": false,
            "allowBackHere": false,
            "previousStep": "FINAL_QUOTE",
            "stepName": "CONFIRMED_BOOKING",
            "created": "2022-07-21T03:02:17.176042Z",
            "modified": "2022-07-21T03:02:17.176043Z"
        },
        {
            "id": 472288,
            "status": "DONE",
            "milestone": "FINAL_QUOTE",
            "caption": "Submit passenger details, accept final quote, and confirm your booking.",
            "sequenceNumber": 2,
            "previousStepSequenceNumber": 1,
            "nextStepSequenceNumber": 3,
            "nextStepConfirmedBooking": true,
            "nextStepTemporaryHold": false,
            "allowBackHere": true,
            "previousStep": "FARE_SELECTION",
            "stepName": "FINAL_QUOTE",
            "expires": "2022-07-21T03:16:30.470045Z",
            "created": "2022-07-21T03:01:30.521054Z",
            "modified": "2022-07-21T03:02:17.178037Z",
            "questions": {...},
            "answers": {...},
            "finalQuote": {...}
        },
        {
            "id": 472287,
            "status": "DONE",
            "milestone": "FARE_SELECTION",
            "caption": "Select product fares.",
            "sequenceNumber": 1,
            "nextStepSequenceNumber": 2,
            "nextStepConfirmedBooking": false,
            "nextStepTemporaryHold": false,
            "allowBackHere": true,
            "stepName": "FARE_SELECTION",
            "created": "2022-07-21T03:01:22.420575Z",
            "modified": "2022-07-21T03:01:30.532235Z",
            "fareDetails": {...},
            "fareSelections": [...]
        }
    ],
    "roadmap": "FARE_SELECTION,FINAL_QUOTE,CONFIRMED_BOOKING",
    "bookings": [...],
    "product": {...},
    "simulation": true,
    "directConnect": true,
    "billingInfo": {...},
    "answers": {},
    "errors": []
}
```

When you POST a request to go back, the system follows the `previousStepSequenceNumber` of the current `ACTIVE` step and takes the flow back to the step with that `sequenceNumber`, reverting it from `status:DONE` to `ACTIVE`. The abandoned steps are preserved in our system but will no longer appear in the flow published on the API.

3.3.4 Error Handling

Of course, sadly things don’t always go to plan. When an active step encounters an error because input required from the customer was not supplied or didn’t pass validation by the Livn API or the reservation system, or because of any other issue while communicating with the reservation system, the response will be a flow with the current step marked with `status:FAILED` and with the details of the underlying issue in `step.error`.

```
{
    "id": 472299,
    "status": "FAILED",
    "milestone": "FINAL_QUOTE",
    "caption": "Submit passenger details, accept final quote, and confirm your booking.",
    "sequenceNumber": 2,
    "previousStepSequenceNumber": 1,
    "nextStepSequenceNumber": 3,
    "nextStepConfirmedBooking": true,
    "nextStepTemporaryHold": false,
    "allowBackHere": true,
    "previousStep": "FARE_SELECTION",
    "stepName": "FINAL_QUOTE",
    "expires": "2022-07-21T03:34:44.510589Z",
    "created": "2022-07-21T03:19:44.539095Z",
    "modified": "2022-07-21T03:20:00.071357Z",
    "questions": {
        "questionGroups": [
            {
                "caption": "Passenger Details - STANDARD - Passenger 1",
                "questions": [
                    {
                        "uuid": "00000001-0019-0007-0002-000000000000",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "PAX_FIRST_NAME",
                        "title": "First Name",
                        "example": "John",
                        "required": true
                    },
                    {
                        "uuid": "00000001-0019-0007-0002-000000000001",
                        "fareUuids": [
                            "00000001-0019-0007-0001-000000000000"
                        ],
                        "scope": "PER_GROUP",
                        "answerType": "TEXT",
                        "purpose": "PAX_LAST_NAME",
                        "title": "Last Name",
                        "example": "Smith",
                        "required": true
                    },
                    {
```

```
                "uuid": "00000001-0019-0007-0002-000000000002",
                "fareUuids": [
                    "00000001-0019-0007-0001-000000000000"
                ],
                "scope": "PER_GROUP",
                "answerType": "EMAIL",
                "purpose": "PAX_EMAIL",
                "title": "E-mail",
                "example": "john.smith@email.com",
                "required": true
            },
            {
                "uuid": "00000001-0019-0007-0002-000000000003",
                "fareUuids": [
                    "00000001-0019-0007-0001-000000000000"
                ],
                "scope": "PER_GROUP",
                "answerType": "PHONE",
                "purpose": "PAX_PHONE",
                "title": "Phone Number",
                "description": "Phone number",
                "example": "+61491570156",
                "required": true
            },
            {
                "uuid": "00000001-0019-0007-0002-000000000004",
                "fareUuids": [
                    "00000001-0019-0007-0001-000000000000"
                ],
                "scope": "PER_GROUP",
                    "answerType": "DATE",
                    "purpose": "PAX_DOB",
                    "title": "Date of Birth",
                    "description": "Please answer in yyyy-MM-dd format.",
                    "example": "1975-01-25",
                    "required": true
                },
                {
                    "uuid": "00000001-0019-0007-0002-000000000005",
                    "scope": "PER_GROUP",
                    "answerType": "TEXT",
                    "purpose": "SUPPLIER_NOTES",
                    "title": "Comments",
                    "description": "General booking notes or comments you wish to pass to the supplier",
                    "example": "I'm allergic to shellfish",
                    "required": false
                }
            ]
        }
    ]
},
"answers": {
    "answers": [
        {
            "questionUuid": "00000001-0019-0007-0002-000000000000",
            "value": "John"
        },
        {
            "questionUuid": "00000001-0019-0007-0002-000000000001",
            "value": "Smith"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000002",
                "value": "john.smith@email.com"
            },
            {
                "questionUuid": "00000001-0019-0007-0002-000000000003",
                "value": "+61491570156"
            }
        ]
    },
    "finalQuote": {...},
    "error": {
        "internalErrorMessage": "question.uuid=00000001-0019-0007-0002-000000000004, question.title=Date of Birth, error=Required
input not supplied.",
        "customerErrorMessage": "Validation failed: Date of Birth: Required input not supplied.. Please review this step and try
again.",
        "terminateTransaction": false,
        "goBackToRetry": false
    }
}
```

The error element contains messages and instructions on how to proceed with the flow. The internalErrorMessage is intended to help Livn's and our clients' developers when debugging issues and generally includes more technical details specific to the upstream reservation system, while customerErrorMessage should contain a more customer-friendly message.

A terminateTransaction flag with the value true means that unfortunately the flow cannot be recovered and progressed any further, which indicates a severe issue with the product, supplier or reservation system. Lastly, the goBackToRetry flag if true (only possible with terminateTransaction:false) indicates that in order to continue with the flow, the client needs to go back as discussed in the previous subchapter and resume the flow from the step it is reverted to. If both terminateTransaction and goBackToRetry are false, it is possible to make the necessary corrections within the FAILED step and simply resubmit the flow.

3.3.5 Simulated checkout

As mentioned earlier, the API User record created for you will have separate development and production credentials for use in their respective environments.

When using the POST api/flows call in the development environment, you may notice that the flow has a simulation:true flag as with the other examples in this document.

```
{
    "id": 394927,
    "livnReference": "99-394927",
    "productId": 1,
    "date": "2023-07-19",
    "currency": "AUD",
    "steps": [...],
    "roadmap": "FARE_SELECTION,FINAL_QUOTE,CONFIRMED_BOOKING",
    "product": {...},
    "simulation": true,
    "directConnect": true,
    "billingInfo": {
        "billingCurrency": "AUD",
        "billingExchangeRate": 1
    },
    "answers": {},
    "errors": []
}
```

What this means is that while we would otherwise be using live products and communication with the upstream reservation systems, any steps which would incur expenses getting charged to the respective Distributor (`milestone:CONFIRMED_BOOKING`) or consume availability or other limited resources on the supplier's side (`milestone:TEMPORARY_HOLD`) would merely be simulated by the Livn API, which might involve generating dummy booking details and ticket information.

3.3.6 Price Preview

A small deviation from the usual checkout flow is the ability to use the `POST /api/flows/previewPrice` call to request a price preview for a FareSelection and AddOnSelection, i.e. a selection of product fares and optional add-ons in the `milestone:FARE_SELECTION` step, as well as any price impacting answers provided at any later step but prior to reaching the `FINAL_QUOTE` milestone, without actually progressing the flow.

The call's expected request body is the same as you would `POST` to `/api/flows` , but the response is an instance of the quote type containing the total price with currency for display to the customer plus where possible depending on the involved reservation system, a detailed itemised breakdown of the included line items.

3.4 Post-checkout operations

3.4.1 Cancellation

The `DELETE /api/bookings/:bookingId` call allows API clients to cancel confirmed bookings they have previously made by completing a checkout flow. We encourage clients to capture the customer's reason for cancellation and to use the optional request parameter `reason` to allow us to pass this on to the supplier's reservation system.

Upon receipt of this request, the system does two things:

1. Create a cancellation record associated with the booking, i.e. any cancellation attempt is recorded, even in case the cancellation is not possible or fails for any reason.

2. Call the respective connector to carry out the actual cancellation via the upstream reservation system. The response will be a cancellation object as shown below containing the price of cancellation, associated booking details, the reason given in the request and several timestamps that indicate the status of the cancellation. In the sample below, the cancellation has been `confirmed`, meaning the Booking has been successfully cancelled.

```
[
    {
        "id": 13890,
        "flowId": 394927,
        "livnReference": "99-394927",
        "passThroughReference": "99-394927",
        "partyName": "John Smith",
        "partyEmailAddress": "john.smith@email.com",
        "tickets": [...],
        "supplierReference": "0BA99394927/J.Smith",
        "confirmed": "2022-07-21T03:31:02.761255Z",
        "invoice": {...},
        "billingInfo": {
            "billingCurrency": "AUD",
            "billingExchangeRate": 1
        },
        "cancellations": [
            {
                "id": 1118,
                "status": "CANCELLED",
                "created": "2022-07-21T03:32:15.255345Z",
                "modified": "2022-07-21T03:32:15.348176Z",
                "confirmed": "2022-07-21T03:32:15.289724Z",
                "reason": "Cancelling to book another trip"
            }
        ]
    }
]
```

The `status` field indicates the state/outcome of the cancellation request which will further be reflected by these timestamp properties:

- `confirmed`: if present, the cancellation attempt was successful
- `failed`: if present, the cancellation attempt failed
- In the absence of either of the two, the cancellation is still in progress

The API only allows one `IN_PROGRESS` cancellation and will reject further cancellation requests. Once a request reaches the state `CANCELLED` the original Booking has been successfully cancelled with the supplier’s reservation system. `FAILED` cancellations serve as records of requests to cancel, it is possible to request the cancellation again following a `FAILED` attempt.

3.4.2 Other Operations

The API currently includes a series of other operations to retrieve individual flow, bookings, or ticket records by their primary id, as well as the ability to load all records made by the calling API user. The `GET /api/bookings` and `GET /api/tickets` calls can be used to search or fetch records by the different booking references or passenger details.

4. Mock Supplier and Products

The behaviour of any product during the checkout process is controlled by the respective supplier’s reservation system, putting it outside of our direct control. Likewise, products unfortunately may become unavailable at times. This could be a temporary issue, e.g. when the upstream reservation system experiences a planned or unplanned outage or when the supplier decides to pause the product or has yet to extend its operational data to a new season. At other times, products may even become permanently unavailable due to having been discontinued by the supplier.

This degree of product “churn” makes it challenging to provide a consistent and reliable set of products to cover every facet of the Livn API’s large scope of functionality. In fact, we may not even be able to find a workable example of a product that makes use of a particular feature, question/answer type or other fields at any given time.

To overcome this issue and to provide developers with a useful set of as few as possible sample products that between them cover every angle of the Livn API and should prepare an implementation for anything you may expect to later face in production, we have built our so-called mock-up connector.

This service, which otherwise behaves like any of our other connectors, but doesn’t depend on any externally operated reservation system, provides exactly this type of test parkour around a small set of currently only four products. These mock-up products, ids 1, 2, 3 and 4, will be available to any client during development and also form the basis of our certification process, to evaluate integrations and make sure they are ready for production. We recommend that you primarily focus your attention on developing your API integration against these mock-up products before expanding to other products available in the pre-certification catalogue.

5. Certification Process

When you have finished developing and thoroughly testing your client implementation, please contact Livn to initiate the certification process.

We will need you to perform a series of specific test scenarios that have been outlined in our separate Livn API Certification Guide.
Livn will ask you to use your application in the same way it is intended to operate later in production, including any importing, synchronisation and/or caching operations. The ultimate aim is to demonstrate that the integration is working optimally and is capable of handling all facets of the search and checkout process.

Where possible, we welcome the opportunity to get access to the customer-facing interface of your application during certification to play through the test cases ourselves, which combined with our server logs should give our engineers a pretty good picture of whether or not your application is ready to go into production or to make recommendations on improvements or necessary critical changes.

Once certified, you will be issued with production credentials and the details of the API endpoint server or servers they have been enabled for.

Important: Please refrain from making live test bookings using a production endpoint and credentials without previous consultation with both the Supplier and Livn. Any cost incurred through bookings made on the live system, which cannot be cancelled, will have to be invoiced to your account.