

# Livn API v2.0 Developer Guide

Revision Date: 2021-05-18

## Table of contents

[Table of contents](#)

[1. About Livn and the Livn API v2](#)

[2. Definition of Terms](#)

[Add On](#)

[Answer](#)

[Base Variant](#)

[Booking](#)

[Billing Info](#)

[Cancellation](#)

[Cancellation Policy](#)

[Catalogue](#)

[Category](#)

[Channel](#)

[Company](#)

[Credential Set](#)

[Departure](#)

[Distributor](#)

[External Resource](#)

[Fare](#)

[Flow](#)

[From Price](#)

[Invoice](#)

[Line Item](#)

[Pax](#)

[Price](#)

[Price Preview](#)

[Product](#)

[Question](#)

[Quote](#)

[Reservation System](#)

[Select Option](#)

[Seller](#)

[Step](#)

[Supplier](#)

[Ticket](#)

[Time Slot](#)

### [3. API Overview](#)

#### [3.1 Security](#)

[3.1.1 Basic HTTP authentication](#)

[3.1.2 JSON Web Tokens \(JWT\)](#)

#### [3.2 Product Search & Discovery](#)

#### [3.3 Wizard-style Checkout a.k.a. The Flow](#)

[3.3.1 The Happy Path](#)

[3.3.2 Moving backwards against the flow](#)

[3.3.3 Error Handling](#)

[3.3.4 Simulated checkout](#)

[3.3.5 Price Preview](#)

#### [3.4 Post-checkout operations](#)

[3.4.1 Cancellation](#)

[3.4.2 Other Operations](#)

### [4. Certification Process](#)

# 1. About Livn and the Livn API v2

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

Clients who integrate the Livn API, from large GDSs and OTAs to virtually any interested travel retailer, can gain easy access to the global market of tours & activities with real-time availability, rates, as well as curated and enriched content, and are able to make instantly confirmed reservations through a single unified 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 by 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 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](http://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 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 for as long as there have been travel agents. 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, 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 which will fundamentally dictate the complexity and the number and sequence of steps towards making a confirmed reservation. Modelled on the sales process conducted by a living and breathing travel consultant in particular, but ultimately any salesperson selling arbitrary goods or services, 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.



Possibly the biggest difficulty with API aggregation and any attempt to try and unify more than a few different API models into just one 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 regards 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 is 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 types of products.

However, the ultra-flexible nature of the checkout stage, needed to overcome the combined complexities of eventually dozens of different source reservation systems, is in large parts focussing 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.

## 2. Definition of Terms

For 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 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 applies). The *billingCurrency* is the currency in which Livn is billing a non-DC retail Channel's Seller buying from wholesale, or where direct connect customers are billing the Channel's Distributor. The *billingExchangeRate* is the exchange rate between the *billingCurrency* and the *Flow.currency*.

## Cancellation

All details of a cancellation or cancellation attempt 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, that 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, maintaining the cache of availability and pricing and of course to interact with the reservation system and writing 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

A 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 served 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.

## 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 which 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, that 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](mailto: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 with, 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.

Clients can use the calls on the `/api/suppliers` and `/api/products` paths to fetch the list of Suppliers and each Supplier's Products available to them. This approach is recommended for integrations that wish to cache parts of the inventory and possibly also perform all search operations on their end.

The alternative and more commonly used approach is to begin every sales process 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`.

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

The Livn API v2 Flow is the term used for both the sales session record and process which starts after the customer has chosen a product and departure/attendance date they wish to purchase, with the aim to complete with 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 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.

The Flow will be returned with a newly assigned id, a roadmap offering a guide to which steps can 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.

```

{
  "id":21,
  "livnReference":"5005-21",
  "productId":5,
  "date":"2021-11-15",
  "currency":"GBP",
  "steps":[
    {
      "id":60,
      "status":"ACTIVE",
      "milestone":"FARE_SELECTION",
      "caption":"Select product fares and add-ons.",
      "sequenceNumber":1,
      "nextStepSequenceNumber":2,
      "allowBackHere":true,
      "stepName":"FARE_SELECTION",
      "created":"2021-04-06T15:03:21.4649537+10:00",
      "fareDetails":{"
    }
  ],
  "roadmap":["FARE_SELECTION",[1],FINAL_QUOTE,CONFIRMED_BOOKING"],
  "product":{"
    "id":5,
    "catalogueProductId":5,
    "disabled":false,
    "supplier":{"
      "id":5,
      "catalogueSupplierId":5,
      "disabled":false,
      "name":"Golden Tours",
      "nameCompany":"Gray Line London",
      "nameTradingAs":"Golden Tours",
      "businessNumber":"2608886",
      "description":"Golden Tours is London's foremost sightseeing company.\nEstablis
      "website":"https://www.goldentours.com/",
      "tnc":"https://dev1.livngds.com/livngds/api/operators/196/tnc",
      "address1":"Office 103 58-60 Petty France",
      "city":"London",
      "postcode":"SW1H 9EU",
      "country":"GB",
      "email":"denisa@goldentours.com",
      "demo":false,
      "v1Cid":893,
      "created":"2021-03-18T13:06:42.955484Z",
      "customMeta":null,
      "resSystem":"LIVNV1"
    },
    "catalogue":{"
      "id":5,
      "name":"Livn Default Catalogue-2021-03-19T11:47:10.444964600+11:00"
    }
  },
  "distributor":{"
    "id":5,
    "name":"Testdistributor2021-03-19T11:47:10.396971600+11:00",
    "company":{"
      "id":1,
      "code":"TestLivnv1",
      "name":"Test Livnv1 company",
      "nameCompany":"Test Livnv1",
      "nameTradingAs":"TSTL",
      "businessNumber":"1231232131356",
      "address1":"402/410 Elizabeth Street, Surry Hills, Sydney, 2010, NSW",
      "city":"Sydney",
      "state":"NSW",
      "postcode":"2010",
      "country":"AU",
      "tz":"Sydney/Australia",
      "email":"test@gmail.com",
      "phone":"02130123012",
      "created":"2021-03-14T15:22:53.316Z",
      "modified":"2021-03-14T15:22:53.316Z"
    },
    "tnc":"Terms And Conditions",
    "emailRes":"res@google.com",
    "emailCancellations":"cancelstions@google.com",
    "customMeta":null
  },
  "channel":{"
    "id":5,
    "directConnect":true,
    "billingCurrency":"USD"
  },
  "currencies":[
    "GBP"
  ],
  "name":"2D Edinburgh Tour ex London",
  "nameOriginal":"2D Edinburgh Tour ex London",
  "description":"Visit Edinburgh and be charmed by the lush beauty and sheer Natur
  "pickupNotes":"Make your own way to the departure point:\nKing's Cross Station w
  "dropoffNotes":"Estimated arrival time back in London is around 22:18.",
  "specialNotes":"Conditions:\nYou will be provided accommodation in one of the th
  "groupSizeMax":10,
  "duration":172800000,
  "fromPrices":[
    {
      "amount":312,
      "currency":"GBP"
    }
  ],
  "operatingDays":31,
  "demo":false,
  "v1Cid":53701,
  "created":"2021-03-18T13:10:26.783025Z",
  "commissionPerc":30,
  "resSystem":"LIVNV1",
  "usesNetRates":false,
  "customMeta":null
},
  "simulation":false,
  "directConnect":true,
  "billingInfo":{"
    "billingCurrency":"USD",
    "billingExchangeRate":1.3892
  }
}

```

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 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.

- 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.

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 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. Hence this type of step is not very common and would generally only be used to present the customer with a final Quote that they must accept before proceeding to write a reservation, while there is no further information left to collect which has no impact on the quoted price, e.g. pax details.

### FareDetails and Selections

```

"steps": [{
  "id": 47,
  "status": "ACTIVE",
  "milestone": "FARE_SELECTION",
  "fareSelections": [{
    "uuid": "00000550-0002-0010-0001-000000000007",
    "quantity": 2
  }],
  "addOnSelections": [{
    "uuid": "00000550-0002-0010-0001-000000000002",
    "quantity": 1
  }, {
    "uuid": "00000550-0002-0010-0001-000000000003",
    "quantity": 1
  }],
  "fareDetails": {
    "baseVariants": [{
      "name": "Standard - Standard",
      "timeSlots": [{
        "name": "10:00",
        "timeStart": "10:00:00",
        "fares": [{
          "uuid": "00000550-0002-0010-0001-000000000007",
          "name": "Adult",
          "unitsAvailable": 10,
          "price": {
            "amount": 266,
            "currency": "AUD"
          },
          "otherCharges": "AUD 7.0: A Government Environmental Management Charge of $6.50 is payable by each adult and each child (4 years and over) entering the Great Barrier Reef World Heritage area. These charges are included in the total price shown.",
          "addOns": [{
            "uuid": "00000550-0002-0010-0001-000000000002",
            "name": "Advanced Guided Snorkel Tour",
            "description": "Passengers under 12 years of age must be accompanied by an adult.",
            "unitsLinkedToParent": false,
            "price": {
              "amount": 87,
              "currency": "AUD"
            }
          }, {
            "uuid": "00000550-0002-0010-0001-000000000003",
            "name": "Introductory Guided Snorkel Tour",
            "description": "Passengers under 12 years of age must be accompanied by an adult.",
            "unitsLinkedToParent": false,
            "price": {
              "amount": 68,
              "currency": "AUD"
            }
          }
        ]
      }
    ]
  }
}

```

### Questions and Answers

```

"steps": [
  {
    "id": 27,
    "status": "ACTIVE",
    "milestone": "FARE_SELECTION",
    "caption": "Select product fares and add-ons.",
    "sequenceNumber": 1,
    "nextStepSequenceNumber": 2,
    "allowBackHere": true,
    "nextMilestone": "PASSENGER_DETAILS",
    "created": "2021-01-27T12:54:54.389234+11:00",
    "modified": "2021-01-27T12:55:51.9442504+11:00",
    "answers": [
      {
        "questionUuid": "00000002-0030-0001-0001-000000000082",
        "value": "00000002-0030-0001-0001-000000000084",
        "hint": "Pick up/ Drop off questions. > Pick up options"
      },
      {
        "questionUuid": "00000002-0030-0001-0001-000000000090",
        "value": "00000002-0030-0001-0001-000000000092"
      }
    ]
  },
  {
    "questions": {
      "questionGroups": [
        {
          "caption": "Pick up/ Drop off questions.",
          "questions": [
            {
              "uuid": "00000002-0030-0001-0001-000000000082",
              "fareUuids": [
                "00000002-0030-0001-0001-000000000005",
                "00000002-0030-0001-0001-000000000006"
              ],
              "scope": "PER_UNIT",
              "answerType": "SELECT_SINGLE",
              "title": "Pick up options",
              "description": "Pick up options",
              "selectOptions": [
                {
                  "uuid": "00000002-0030-0001-0001-000000000083",
                  "title": "I don't need a pick-up, I will meet you on location.",
                  "description": "I don't need a pick-up, I will meet you on location."
                },
                {
                  "uuid": "00000002-0030-0001-0001-000000000084",
                  "title": "Airport",
                  "description": "Sunshine Coast Airport"
                }
              ]
            }
          ]
        }
      ]
    }
  }
]

```



```
    ]
  }
  ],
  "available": true
}
],
"available": true
}
]
}
```

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` (if 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 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.

```

"steps": [
  {
    "id": 16,
    "status": "ACTIVE",
    "milestone": "PASSENGER_DETAILS",
    "caption": "Submit passenger and booking details.",
    "sequenceNumber": 2,
    "previousStepSequenceNumber": 1,
    "nextStepSequenceNumber": 3,
    "allowBackHere": true,
    "previousMilestone": "FARE_SELECTION",
    "nextMilestone": "FINAL_QUOTE",
    "expires": "2021-03-06T00:16:57.4352667Z",
    "created": "2021-01-05T11:16:57.4602662+11:00",
    "questions": { }
  },
  {
    "id": 15,
    "status": "DONE",
    "milestone": "FARE_SELECTION",
    "caption": "Select product fares and add-ons.",
    "sequenceNumber": 1,
    "nextStepSequenceNumber": 2,
    "allowBackHere": true,
    "nextMilestone": "PASSENGER_DETAILS",
    "created": "2021-01-05T00:04:39.040783Z",
    "modified": "2021-01-05T11:16:57.4602662+11:00",
    "questions": { },
    "fareDetails": { },
    "fareSelections": [ ],
    "answers": { }
  }
}

```

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.

### 3.3.3 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":14,
  "status":"FAILED",
  "milestone":"FARE_SELECTION",
  "caption":"Select product fares and add-ons.",
  "sequenceNumber":1,
  "nextStepSequenceNumber":2,
  "allowBackHere":true,
  "nextMilestone":"PASSENGER_DETAILS",
  "created":"2021-01-04T16:34:21.764581+11:00",
  "modified":"2021-01-05T13:06:27.2457334+11:00",
  "questions":{ },
  "fareDetails":{ },
  "error":{
    "internalErrorMessage":"Fare selection is required",
    "customerErrorMessage":"Validation failed: Fare selection is required. 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 might include technical details specific to the upstream reservation system, while `customerErrorMessage` contains a customer-friendly message.

A `terminateTransaction` flag with value `true` means that unfortunately the flow cannot be processed 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 in the FAILED step and resubmit the flow.

### 3.3.4 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.

```
{
  "id":16,
  "referenceId":"5005-16",
  "productId":2,
  "date":"2021-01-25",
  "currency":"PHP",
  "steps":[ ],
  "roadmap":"FARE_SELECTION,PASSENGER_DETAILS,FINAL_QUOTE,TEMPORARY_HOLD,CONFIRMED_BOOKING",
  "product":{" },
  "simulation":true,
  "directConnect":false
}
```

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 (e.g. `milestone:CONFIRMED_BOOKING`) or consume availability or other limited resources on the supplier's side (e.g. `milestone:TEMPORARY_HOLD`) would merely be simulated by the Livn API, including generating dummy booking details and ticket information.

### 3.3.5 Price Preview

A small deviation to 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 without actually progressing the flow.

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

## 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": 2,
  "created": "2021-01-05T04:49:42.2328013Z",
  "confirmed": "2021-01-05T04:49:42.2547948Z",
  "booking": {
    "id": 2,
    "tickets": [
    ],
    "supplierReference": "5005-18",
    "confirmed": "2021-01-05T15:48:06.185593+11:00",
    "externalResources": {
    },
    "invoiceDetails": {
    },
    "cancellationPolicy": {
    }
  },
  "price": {
    "amount": 0,
    "currency": "PHP"
  },
  "reason": "Cancelling this booking due to change of plans."
}

```

The status of the cancellation request will be reflected by these variables:

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

The API will only allow clients to perform one cancellation, regardless of an existing cancellation attempt's status.

### 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. Certification Process

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

Livn will ask you to use your application in the same way it is intended to be used in production, including any importing, synchronisation and/or caching operations. We would then request that you perform a series of test booking and cancellations of specific products and dates to demonstrate that the integration is working optimally and is capable of handling all facets of the search and checkout process.

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 be invoiced to your account.