viator | Partners

Technical Guide for API Partners

Welcome to the Viator Partner Program! In this guide, we'll go through a variety of topics regarding how you can best use the Viator Partner API to display, promote and sell Viator's 300,000+ products on your own website or travel-booking system. For our partners who use the booking functionality we will also discuss how to interact with our booking systems via the Partner API to support your booking/checkout workflow.

All **documentation** pertaining to the Viator Partner API can be found here. If you have any questions about your integration, please reach out to affiliateapi@tripadvisor.com at any time.

Please note that which endpoints you are able to access depends on what kind of partner you are – merchant or affiliate (with basic, full, or booking access). Different endpoint access levels can be verified here. This guide includes information about the endpoint access for each partner type. For example, endpoints marked as below would not be accessible by Basic-access Affiliate:

⊗ Basic-access Affiliates **✓ Full-access Affiliates** **✓ Merchants**

Furthermore, in a small number of cases an endpoint accessible to both partner types will behave differently between merchants and affiliates. We will mention these differences in the relevant sections.

Getting started: choosing ingestion-model or search-model

As the **first implementation step**, you will need to review the API documentation – section **"Product content and availability endpoints"** and decide which model for data management you wish to support.

We offer two models for data management:

Ingestion model

- **data ingestion with modified-since endpoints:** /products/modified-since for product content and /availability/schedules/modified-since for availability and pricing schedules (**at least hourly**).
- The ingestion model of connecting to Viator's supply requires that the partner builds and maintains a **local database** of products and also requires that the partner makes regular calls to Viator to ensure that product and availability information is up to date.
- This model allows an **easy ingestion** of all Viator product content and availability data into partner's database with the possibility to **manage and filter** products according to the business needs.
- **This model is ideal for partners that want a full-scale tours and activities integration** and can manage large amounts of data.

Real-time search model

- **real-time search** for products with the **search endpoint** (/products/search or /search/freetext) and getting product details with a **single product content** (/products/{product-code} and **single availability** (/availability/schedules/{product-code}) endpoints used in **real-time** to check the information for only **one product** selected by the customer from the search results.
- The search model is an **easier and faster** way to connect to Viator's supply via an API integration. The amount of data returned in a search API request is less than what would be returned using the ingestion model, and

therefore there is no need to store any data. Partners can benefit from pre-built search functionalities available in the Viator's search endpoints.
- **This model is ideal for partners that are looking for a fast integration** and do not desire a full tours and activities integration.

Important:

The /bulk endpoints (/products/bulk and /availability/schedules/bulk) **must not be used** for ingestion or in real-time, they could be used only for some **edge cases** when you need to get details of a few products at the time. Please refer to our API documentation (here).
Both supported models are described in this article: **Managing product and availability data**.

The models described above refer to managing the product content and availability. In addition to that, it's necessary to comply with the **requirements for endpoint usage applicable to auxiliary data** such as locations, reviews, tags, etc. All auxiliary data must be cached and refreshed based on the frequency specified for each endpoint in the API documentation – section **"Update frequency"**.

Quick find

Attractions

- Retrieve all attractions
- Get details of a single attraction

Important validations – product options

- Product option title and description displayed
- Language guides validated
- Pickup / drop-off verified for product options
- Finding meeting points

Refreshing your data quick guide

- Fixed-cadence delta updates
- Fixed-cadence full updates
- On-demand updates

Building your local database – taxonomy

All our products are categorized under a destination hierarchy. You will use this hierarchy to build your database and effectively catalogue our products.

/destinations

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Get details of all destinations

- Every product in the Viator catalogue is categorized according to the destination/locale in which it operates. This endpoint returns a complete list of Viator destinations, including destination names and parent identifiers (for the hierarchy of destinations check the lookupId field).
- This endpoint is used to help define the destinations of products and attractions (i.e. primary and secondary) for the purpose of merchandising products. For example, Disneyland Paris's primary destination is Paris, even though the actual destination (secondary destination) is Marne-la-Vallee. Travelers are more likely to search for Paris when looking for Disneyland Paris than Marne-la-Valle.
- This endpoint is used to provide navigation on your site, through drilldown lists, combo boxes, or breadcrumbs.
- Fields returned in the response can be used to map a destination to an IATA code (airlines) or language.
- Destination data must be **cached**. While destinations rarely change, we recommend refreshing the list of destinations **weekly**.

You can also create a catalogue of attractions and merchandize products based on the attraction they are linked to.

/attractions/search

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Get all attractions for a destination

- This endpoint returns **all attractions within a destination** (facilitates mapping destinations to attractions) and is used to help merchandize products by attraction.

- Attractions can be **sorted** by ALPHABETICAL, DEFAULT & REVIEW_AVG_RATING order.
- **Pagination rules apply** to this endpoint: no more than 30 products can be requested in a single call ("count": 30). The first request should be done with "start": 1, the second request must include the value of the start parameter for the next result you wish to see. For example, if the first request was done with "start": 1 and "count": 30, the second request will be done with "start": 31 and "count": 30, the third one with "start": 61 and "count": 30 etc.
- **Attraction details** include: destination mapping, Viator attraction URL (Affiliate partners only), number of products + product codes, reviews, images, introduction, overview, details of admission type (free/not free), opening hours, full address.
- When a product is linked to an attraction the relevant attractionId is returned for a product in the response to the product content endpoint. You can **map products to attractions** using the attractionId extracted from the product content response.
- This is also useful for **navigation** as well as **building out basic attractions pages**.
- With this endpoint you can get aggregated **product and review count/rating** for an attraction.
- Attraction data **must be cached** and refreshed **weekly**.

↑ Back to top

Managing product content

There are three product content endpoints that are used to get product data. We highly recommend ingesting all product content into your database. By using a local database, you need only perform a single initial ingestion of data; then, only new and updated product content will be ingested. This will result in faster load times and will overall provide a better experience for travelers. If ingestion is not an option for you, you can request the details of a single product making a real-time call when the customer selects the product on your site.

Key features of these endpoints

Ingest over 300,000+ products quickly and efficiently

Updating product content with fewer, but more frequent requests **provides travelers with accurate information**

Only ingesting new or modified product information **speeds up load times**

Structured data fields make it easier to **break out key information and merchandise products**

Ingest products & keep the product catalogue up to date

/products/modified-since

⊗ **Basic-access Affiliates** ✓ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

1a. Perform an initial ingestion of all product data

- Performs an initial ingestion of all product data into a local database. You may filter out any products during the initial ingestion. You may filter out on your end any products based on the response to this endpoint.
- For the initial ingestion of the product catalogue, the first request to /products/modified-since must include only the count parameter in the request body (value not higher than 500). This way you will start ingesting all Viator products from scratch. This is the only time when the cursor or the modified-since parameter should not be used. The nextCursor parameter returned in the response points to the next page of results and must be used the next call to /products/modified-since. This process must be repeated until the nextCursor is not returned in the response – this means that there are no more results.

1b. Ingest only new or updated product data

- Used to ingest only new or updated product information, as well as to identify deactivated products. Products are considered updated when the supplier makes any changes to the product's details, excluding pricing and availability changes, which are retrieved from the availability schedule endpoint. This endpoint returns all product details (not only changes) for updated products and all details must be refreshed in your database accordingly.
- We recommend polling this service every 15 minutes to look for updates and to avoid any discrepancies. **Updates must be ingested at least hourly**.
- **Method**: Continuously poll all updates using the new nextCursor value until no more updates are left.

- The modified-since parameter should not be used as the standard pagination method; rather, **use the cursor parameter to ensure all product updates are captured during ingestion.** The valid use case for modified-since parameter is to re-sync data if there was an issue on your side and you need to re-ingest updates from a specific time in the past, i.e. yesterday. In such cases you could re-start the job using the last crawled date as the modified-since date on first request and then continue ingesting updates with the cursor parameter. You should never use both the cursor and the modified-since parameters in the same request.
- For **pulling updates**, remember to use the cursor parameter set to the value of the most recent nextCursor element returned by a request to the same endpoint. Then continue to make requests with the cursor parameter set to the value of the nextCursor element of the previous request, until a request again does not include a nextCursor value.
- You need to track the last nextCursor you received as this will be used to update the product catalogue during the next ingestion.
- You should **never re-ingest the entire product catalogue** unless you need to re-initialize your database.

Important: Only this endpoint can be used to ingest the product catalog. You can read more about supported models for managing the product and availability data in this article: **Managing product and availability data**.

Get details of a single product in real-time

/products/{product-code}

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

2. Pull product data for a single product in real-time

- Pulls info for a single product by providing the corresponding product code.
- Used in **real-time** to get details of a **single product when the customer selects a product from the search results** returned with the search

endpoint (/products/search or /search/freetext).

- The response can be cached for up to 1 hour but this endpoint must not be used to ingest content for all products.
- This endpoint must not be used if you're already ingesting products with the /products/modified-since endpoint as both endpoints pull data from the same source. The /products/{product-code} endpoint doesn't return more accurate data but it's used in real-time to fetch product information when needed if the product details haven't been ingested into partner's database with the /products/modified-since endpoint.

Pull details of selected products if needed

/products/bulk

⊗ **Basic-access Affiliates** ✓ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

3. Pull product data for multiple products

- This endpoint pulls in product information for a specified **list of products**, up to 500 products per request.
- **Special use cases:**
  1. **Ingestion errors:** in the case that products were not correctly ingested via the /products/modified-since endpoint or if for any other reason you need to need to fix some products on your end, you can use /products/bulk to reingest those products. This use case applies only to a situation when specific products with known product codes haven't been ingested correctly on your end, not to a situation when it's necessary to re-ingest product updates for all products from a specific date in the past – for that you must use the /products/modified-since endpoint.
  2. **Ingestion of <10k products**: this endpoint maybe used as part of a regular ingestion process only if you have a small curated product list (<10k) that you want to refresh on schedule. If this doesn't apply to your implementation and you need to ingest product content for many / all products, you must use the /products/modified-since endpoint for this purpose.

3. **Recommendations**: using this endpoint to fetch details of a few products when generating similar product recommendations with the /products/recommendations endpoint – see section Enhancing product discovery.

What's included in the response to product content endpoints?

These endpoints return all product information, including, but not limited to:

- Titles
- Descriptions
- Ticket types
- Supplier photos*
- Review ratings and counts
- Meeting points
- Traveler pick up details (if applicable)

- Inclusions and exclusions
- Cancellation policy
- Additional information from the supplier
- Booking confirmation settings (e.g. instant confirmation or on-request products)
- Booking requirements (e.g. min or max travelers)

- Booking questions
- Itineraries
- Product options
- Supplier name

*Viator also allows travelers to upload their own photos. Traveler photos will have to be ingested via /reviews/product.

Creating the search functionality with search endpoints

/products/search or /search/freetext

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Pull product summaries based on search criteria

- When product summaries are returned, they contain a small amount of **crucial product information** such as, but not limited to, title, short descriptions, images, pricing, and review ratings and counts.
- You can effectively filter out products returned in the request by **specifying search criteria**, such as destination IDs, price range, and date range. You also have the option to filter by category using tag IDs and can filter by flags, such as free cancellation or skip-the-line tours.
- The search functionalities help **identify high-quality products** using tags (applicable only to the /products/search endpoint; see the quality-related tags in this article: Viator tags, explained) or flags ("LIKELY_TO_SELL_OUT" – popular products that routinely sell out)
- The /search/freetext endpoint allows an easy implementation of the search functionality based on the desired **search term** (supported search types: ATTRACTIONS, DESTINATIONS, PRODUCTS).
- You will specify the **format** in which you want the response, such as the language and currency and will specify how the response will be **sorted**. You can apply the "featured" ("DEFAULT") sort order from Viator to display featured products first in search results. *Note: Tours, activities and experiences are ranked using exclusive Viator data that may include product quality, reviews, ratings, photos, popularity, user preferences, price, bookings made through Viator, and payments made by operators.*
- These endpoints are ideal for where **short product summaries** would be needed, such as on search cards on search results pages.
- **Pagination rules** apply to these endpoints: No more than 50 products can be requested in a single call ("count": 50). The first request should be done

with “start”: 1, the second request must include the value of the start parameter for the next result you wish to see. For example, if the first request was done with “start”: 1 and “count”: 50, the second request will be done with “start”: 51 and “count”: 50, the third one with “start”: 101 and “count”: 50 etc.
You should paginate through the search results (using the start and the count paremeters) only when the customer wants to move to the next page with search results to see more products. First, you can retrieve and display maximum 50 products and if the customer wants to move to the next page with search results, another request to the search endpoint should be done to request additional products. **You shouldn’t pull automatically all products from each destination when the user initiates the search**, this would be an incorrect usage of the endpoint and would result in long load times.

- The search endpoints **must not be used for ingestion**, the /products/modified-since endpoint must be used for that purpose. These endpoints can be used only for **real-time searches** (results may be cached for up to 1 hour, anything above that would risk stale data).

Displaying location details

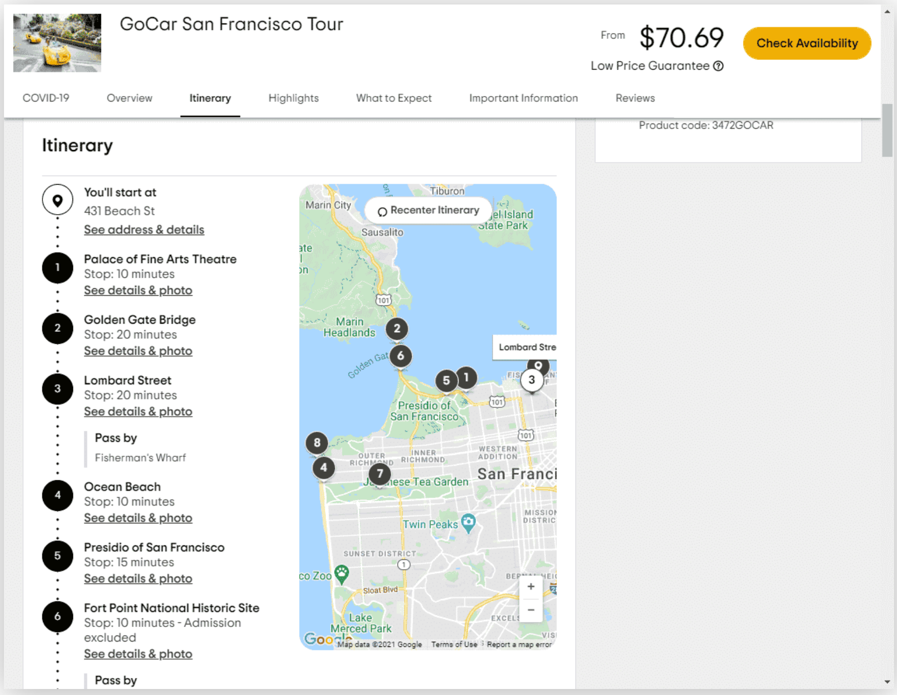
/locations/bulk

✓ Basic-access Affiliates **✓ Full-access Affiliates** **✓ Full + Booking access Affiliates** **✓ Merchants**

Get full location details for the requested location references

- **Location details** returned in /locations/bulk include the name of the location, the street address, and the longitude/latitude coordinates of the location. The locations are not to be confused with destinations retrieved with the /destinations endpoint.

- Locations details can be used to **highlight pickup/drop off points** or **meeting point locations**, to **build itineraries**, or even **overlay locations on a map** to help provide additional context to what a traveler can expect on the tour.
- This endpoint can be used to provide **pickup locations** to travelers through a drop-down list. In addition to that, travelers can specify custom pickup info through a plain text field (when a custom pickup location is supported based on the value returned in the "allowCustomTravelerPickup" field).
- There is a lot of location data in the API and it doesn't change frequently. To avoid too many unnecessary requests to this service, **location data must be cached** and it should be refreshed monthly. Additionally, the /locations/bulk endpoint should be used **on demand** for any new location references returned in the product content response.



Itinerary

You'll start at
431 Beach St
See address & details

1 Palace of Fine Arts Theatre
Stop: 10 minutes
See details & photo

2 Golden Gate Bridge
Stop: 20 minutes
See details & photo

3 Lombard Street
Stop: 20 minutes
See details & photo

Pass by
Fisherman's Wharf

4 Ocean Beach
Stop: 10 minutes
See details & photo

5 Presidio of San Francisco
Stop: 15 minutes
See details & photo

6 Fort Point National Historic Site
Stop: 10 minutes - Admission excluded
See details & photo

Pass by

Example of a structured itinerary with a map overlay

Categorizing products and promoting high-quality products using tags

/products/tags

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

1. Ingest tags into a local database with /products/tags

- Tags should be **cached** and refreshed **weekly**.

2. Build a hierarchical structure of categories and subcategories

- Tag taxonomy allows an easy way to **classify products** into categories and subcategories.
- Each tag can have one or more parents – identified by parentTagId in the API response. This way tags can be organized in a **hierarchical way** with main categories and subcategories.

3. Create custom filtering options

- Filtering by tags is available as a pre-built solution in the /products/search endpoint to easily identify and display to customers products from relevant categories or products that meet the desired quality level.
  Tags are the easiest way to identify high-quality products that generate the highest sale. These tags are:

tagId 367652 – Top Product
tagId 367653 – Low Supplier Cancellation
tagId 367654 – Low Last Minute Supplier Cancellation
tagId 21972 – Excellent Quality
tagId 22143 – Best Conversion
tagId 22083 – Likely To Sell Out
tagId 11940 – Once in a Lifetime
tagId 21074 – Unique experiences
tagId 6226 – Best Value
tagId 21971 – Viator Plus

4. Work on your merchandising strategy

- Tags can be used for **product curation** and to **create custom marketing campaigns** such as holiday offerings, to help your product and marketing teams maximise the possibilities coming from the API integration with Viator.

Read more about tags in this article: Viator tags, explained.

Enhancing product discovery and recommendations

/products/recommendations

⊗ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Integrate similarity recommendations into your booking flow

- Use the /products/recommendations endpoint through the user journey on your platform to **enhance product discovery** by offering **similar product recommendations**.
- **Implement the recommendations flow correctly:** call the /products/recommendations endpoint to get the product codes for simliar products. Then use the relevant product content endpoint to retrieve the product details as well as the /schedules endpoint to get availability and pricing if needed:

– **Ingestion partners**: Retrieve the product details from the data previously ingested with the /products/modified-since and /availability/schedules/modified-since endpoints.

– **Non-ingestion partners:** Make real-time calls to either the /products/{product-code} or /products/bulk endpoint for product content and the /availability/schedules/{product-code} or /availability/schedules/bulk endpoint for availability and pricing schedules.
Note: If you need to retrieve details for more than 10 products at a time, use the /bulk endpoints. If certain product features are unsupported, and you are unable to sell specific products with those features (e.g., MANUAL

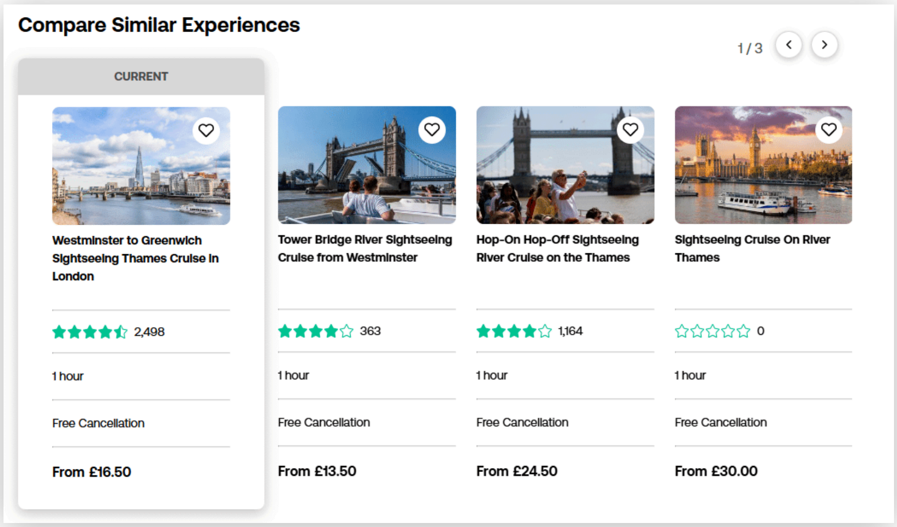
confirmation type), apply filtering to exclude these products based on the product content response.

- **Keep customers engaged** by suggesting similar products, elevating products they may not have seen but would be interested in. This endpoint returns products that are considered similar to the requested product based on machine learning algorithms, considering attributes such as location, category, itinerary, price range and others.
- **Reduce churn** by suggesting alternative products when customers first choice is unavailable.

Key Benefits:

- Algorithm-generated recommendations based on various customer and product metrics.
- Increased sales opportunities through improved product discovery.
- Dynamic recommendation types that can evolve to meet changing business needs.
- Higher customer engagement.
- Improved shopping experience.
- Protection against revenue loss.

Please note: you need to be able to handle cases where no recommendations are available for a particular product.



CURRENT			
Westminster to Greenwich Sightseeing Thames Cruise in London	Tower Bridge River Sightseeing Cruise from Westminster	Hop-On Hop-Off Sightseeing River Cruise on the Thames	Sightseeing Cruise On River Thames
2,498	363	1,164	0
1 hour	1 hour	1 hour	1 hour
Free Cancellation	Free Cancellation	Free Cancellation	Free Cancellation
From £16.50	From £13.50	From £24.50	From £30.00

Example of how similar product recommendations could be implemented.

↑ Back to top

Managing availability and pricing

The Viator Partner API's schedules endpoints enable you to provide availability and pricing information in real-time. To improve transfer speeds and reduce the amount of data transferred, availability is communicated by providing the overall schedule season and specifying unavailable dates rather than available dates. Special pricing periods are also included allowing you to surface supplier promotions to customers.
Ingesting and updating availability schedules follows a similar process as ingesting and updating product information. If ingestion is not an option for you, you can request the schedules of a single product making a real-time call when the customer selects the product on your site.

Key features of these endpoints

Real time availability checks ensure accuracy and **create a seamless checkout**

Only ingesting new or modified availability schedules **speed up load times**

Schedules can be used to **display future availability**

Structured data allows you to easily **build a variety of display and filtering options**

Ingest availability and pricing

/availability/schedules/modified-since

⊗ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

1a. Perform an initial ingestion of all availability and pricing schedules

- Similar to using the /products/modified-since endpoint, you'll use the /availability/schedules/modified-since endpoint to get availability for **all products**. Getting everything at once and updating only new or modified availability schedules will speed up ingestion and ensure availability is not stale.
- You should only ingest schedules for products that are **active** and supported on your platform (filtering must be done on your end). Therefore, schedules should be ingested after product content is ingested.
- For the **initial ingestion** of availability and pricing schedules, the first request to /availability/schedules/modified-since must include only the count parameter in the request body (value not higher than 500). This way you will start ingesting schedules for all Viator products from scratch. This is the only time when the cursor or the modified-since parameter should not be used. The nextCursor parameter returned in the response points to the next page of results and must be used the next call to /availability/schedules/modified-since. This process must be repeated until the nextCursor is not returned in the response – this means that there are no more results.

1b. Make regular calls to check for new or updated availability and pricing schedules

- We recommend polling this service every 15 minutes to look for updates and to avoid any discrepancies. It's essential to conduct these checks **at least every hour** to ensure that your travelers see the most up-to-date pricing and availability. A product's availability is considered modified if a supplier makes changes to pricing or availability.
- **Method**: Continuously **poll all updates** using the new **nextCursor** value until no more updates are left. The modified-since parameter should not be used as the standard pagination method; rather, use the cursor parameter to ensure all schedule updates are captured during ingestion.

The valid use case for modified-since parameter is to re-sync data if there was an issue on your side and you need to re-ingest updates from a specific time in the past, i.e. yesterday. In such cases you could re-start the job using the last crawled date as the modified-since date on the first request and then continue ingesting updates with the cursor parameter. You should never use both the cursor and the modified-since parameters in the same request.

- For **pulling updates**, remember to use the cursor parameter set to the value of the most recent nextCursor element returned by a request to the same endpoint. Then continue to make requests with the cursor parameter set to the value of the nextCursor element of the previous request, until a request again does not include a nextCursor value.
- You need to **track the last nextCursor** you received as this will be used to update the product catalogue during the next ingestion.
- You should **never re-ingest all schedules in full** unless you need to re-initialize your database.

Important: Only this endpoint can be used to ingest the availability and pricing schedules. You can read more about supported models for managing the product and availability data in this article: **Managing product and availability data**.

Get availability and pricing schedules for a single product in real-time

/availability/schedules/{product-code}

✓ Basic-access Affiliates **✓ Full-access Affiliates** **✓ Full + Booking access Affiliates** **✓ Merchants**

2. Pull availability schedules for a single product

- This endpoint returns full availability and pricing information for a **single product**.
- This endpoint can be used in conjunction with /products/{product-code to pull availability and pricing for a single product **when the customer selects**

a product from the search results.

- The response can be cached for up to 1 hour but this endpoint **must not** be used to ingest schedules for all products.
- This endpoint **must not be used** if you're already ingesting schedules with the /availability/schedules/modified-since endpoint as both endpoints pull data from the same source. The /availability/schedules/{product-code} endpoint doesn't return more accurate data but it's used in **real-time** to fetch availability and pricing when needed if the schedules haven't been ingested into partner's database with the /availability/schedules/modified-since endpoint.

Pull schedules for multiple products when needed

/availability/schedules/bulk

⊗ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

3. Pull availability schedules for multiple products

- This endpoint pulls availability and pricing for a **list of products**, up to 500 products per request.
- **Special use cases::**
  1. **Ingestion errors**: in the case that a product schedules were not correctly ingested via the /availability/schedules/modified-since endpoint or if for any other reason you need to need to fix some products on your end, you can use /availability/schedules/bulk to reingest those schedules. This use case applies only to a situation when schedules for specific products with known product codes haven't been ingested correctly on your end, not to a situation when it's necessary to re-ingest schedule updates for all products from a specific date in the past – for that you must use the /availability/schedules/modified-since endpoint.
  2. **Ingestion of <10k products:** this endpoint could be used on a regular basis only if you have a small curated product list (<10k) that you want to refresh on schedule. If this doesn't apply to your implementation and you need to ingest schedules for many / all products, you must use the /availability/schedules/modified-since endpoint for this purpose.

3. **Recommendations**: using this endpoint to fetch details of a few products when generating similar product recommendations with the /products/recommendations endpoint – see section Enhancing product discovery.

Important: The /schedules endpoints return the rates in the **supplier's currency**. You will have to convert the rates to the user's currency either using the /exchange-rates endpoint or your own conversion rates.

Verify real-time availability and pricing

/availability/check

⊗ **Basic-access Affiliates** ✓ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

4. Check availability and price in real-time

- This endpoint enables a **real-time check and calculation of pricing and availability** and should be used throughout the booking path. For example, this endpoint can be used to ensure accuracy when the traveler is starting the checkout process.
- This should not be called until a user inputs **dates and passenger mix**.
- In case of pricing differences between the previously quoted price and the new price from the /availability/check response, the **new price** must be applied to the booking (this shouldn't trigger the booking flow to be canceled, instead the new price should be communicated).
- This endpoint would be used before a booking is made to verify the product is available on the desired date for the desired passenger mix (based on the Viator age bands). **Before a booking hold** can be requested, you'll need to use this endpoint to check that it's still available and make sure the pricing is correct.
- This endpoint **should not be called after requesting the hold**. When there is a valid availability hold for the booking the response to the /availability/check endpoint could return no availability due to a valid

hold. This endpoint could be used again in case the existing hold expires and the booking hasn't been made yet however in that case it would be best to make a new hold right away (up to 3 times).

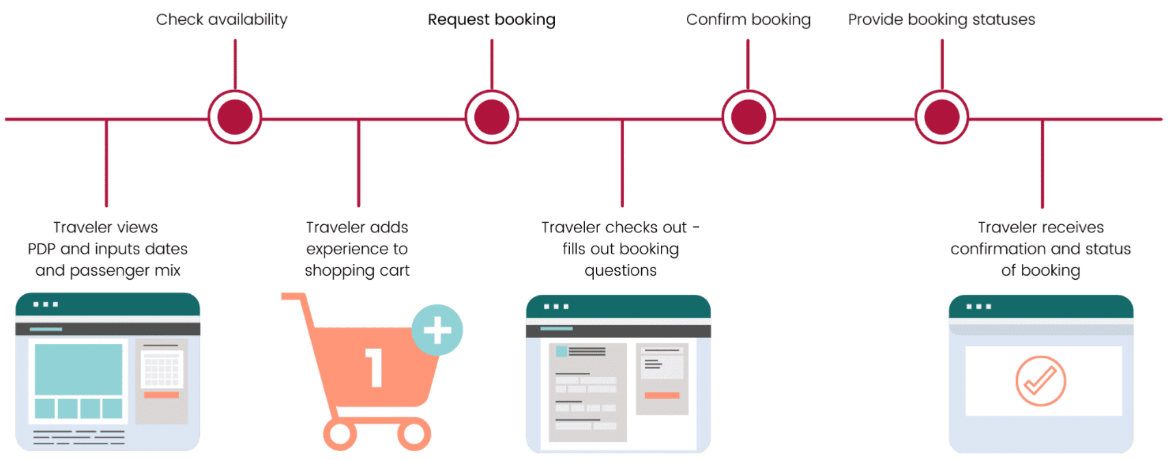
- **If booking hold is not supported in your implementation**, it's essential to call the /availability/check endpoint again in the booking flow, right before submitting the booking request, to double-check the price and availability (even if this check was done already at an earlier stage). Availability and pricing may change at any time, for example while the customer is providing all details for the booking, that's why this information must be verified again before making a booking.
- Full pricing information, including **extra charges paid in destination**, is returned in the response to this endpoint.

↑ Back to top

Creating a seamless booking experience (partners with booking access)

A series of booking endpoints enables you to build booking/checkout user flows that allow travelers to add experiences to shopping carts, provide pickup info, add special requirements (like dietary restrictions), and hold their booking to guarantee availability and pricing.

For Full + Booking affiliates: please see here for more info on how to use our payment solutions to create a booking/checkout experience.


Check availability
Request booking
Confirm booking
Provide booking statuses
Traveler views PDP and inputs dates and passenger mix
Traveler adds experience to shopping cart
Traveler checks out - fills out booking questions
Traveler receives confirmation and status of booking
1

In general, to give your travelers the ability to book a product, you must do the following:

/availability/check

⊗ Basic-access Affiliates	✓ Full-access Affiliates	✓ Full + Booking access Affiliates	✓ Merchants

1. Check availability

- Using traveler inputs of tour date, tour time, and passenger mix, you will perform a real-time check before the traveler is able to add to a shopping cart and/or before placing the booking.

- Follow the rules applicable to the /availability/check endpoint in the previous section.

/bookings/cart/hold or /bookings/hold

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

2. Request a booking hold

- Requests the creation of a booking hold – a guarantee that either the price or availability (or both) of the product will be retained until a booking request is made using the /bookings/cart/book or /bookings/book endpoint.
- Although booking holds are an optional feature, we **recommend using this functionality**. This feature gives travelers flexibility to continue to shop without worrying about any products becoming unavailable.
- **Availability holds** may not be granted, but **pricing holds** will always be granted and is sufficient to confirm bookings.
- The response **confirms the price held**. In case of discrepancy between the held price and the price previously returned in the /availability/check response, the held price from the /hold endpoint applies.
- Full pricing information, including **extra charges paid in destination**, is returned in the response to these endpoints.
- Do not call this service just because the traveler picked a date and passenger mix. Only call this service if a user takes **high-intent actions**, like adding to a cart or during checkout.
- This endpoint should not be called immediately prior to booking to check availability – as for this purpose the /availability/check endpoint should be used (not all products support availability hold therefore availability cannot be verified with the /hold endpoints); it should be called when the customer is moving to checkout or about to provide payment details, to hold availability and pricing for a few minutes.
- **Timestamps** returned for the availability and pricing hold **must be verified** and a new hold should be done in case the first hold expired and the booking hasn't been made yet (we recommend not exceeding 3 holds). There are two timestamps – for availability and for pricing. If the

availability hold expires and the product becomes unavailable before the booking is confirmed, the booking will fail. However if the pricing hold expires and the price changes but availability doesn't change, the booking will be confirmed for the new price (that could be much higher).

- In case the **booking hold is not supported**, it's essential to call the /availability/check endpoint again in the booking flow, right before submitting the booking request, to double-check the price and availability (even if this check was done already at an earlier stage).

/v1/checkoutsessions/{sessionToken}/paymentaccounts or iframe

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ⊗ Merchants

3. Collect payment

This step applies only to Affiliate API partners with booking access who use the **API payments solution** or the **iframe** – see the step-by-step guide. Merchant partners use their own solutions for payment collection.

3a. Payments endpoint: /v1/checkoutsessions/{sessionToken}/paymentaccounts

- Use the paymentDataSubmissionUrl from the /bookings/cart/hold response to **submit customer's card details** for payment.
- Make sure that customer's card information is collected in a secure way following the **PCI compliance requirements**.
- Provide **full card information** in the request to the /paymentaccounts endpoint, including the number, cvv, expiry details, cardholder's full name and billing address. Failure to privde any of this information will result in a payment rejection.
- Implement the Content Security Policy on your site following the requirements from our guide to facilitate fraud detection.

3b. Iframe

- Use the paymentSessionToken from the /bookings/cart/hold response to **initialize the iframe** and call the form submission logic to **process the**

payment details. Use the paymentToken generated in the iframe to **confirm the booking**.

- Make sure that customer's card information is collected in a secure way following the **PCI compliance requirements**.
- Provide **full card information** in the request to the /paymentaccounts endpoint, including the number, cvv, expiry details, cardholder's full name and billing address. Failure to privde any of this information will result in a payment rejection.
- Implement the Content Security Policy on your site following the requirements from our guide to facilitate fraud detection.

See **Affiliate Payments | Viator Partner API guide** for more details.

4. Facilitate fraud prevention

This step applies only to Affiliate API partners with booking access who use the **API payments solution** or the **iframe** – see the step-by-step guide.

The **Viator Javascript library** must be imported into your payment details page using the following URL:
https://checkout-assets.payments.tamg.cloud/stable/v2/payment.js

If using a build tool, for example if you are using webpack to create your pages, and you can import NPM packages then you can use this option to load the library:

https://checkout-assets.payments.tamg.cloud/payment-module-v2.0.4.tgz

Device fingerprinting information must be collected and passed to Viator to facilitate fraud prevention. These data points about the device, connection, and location are then analyzed to recognize trusted customers and reduce the likelihood of fraudulent bookings.
Note: Not providing this information will put your transactions at risk of rejection by Viator's fraud prevention tools.

In addition to fraud prevention measures from the Viator's Javascript library, affiliate API partners with booking access should use additional fields (**FraudPreventionDetails**) available in the API for fraud prevention.
If you work with agencies, agents, or sell through different channels, you should have a way to identify transactions coming from different sources. For

that we have the following fields in the additionalBookingDetails schema in the /bookings/cart/book request in the API:

- subChannelId
- agencyId
- agentId

When it comes to identifying high risk transactions, in some cases it matters who will receive the voucher post booking. For this reason, we also ask you to share this information using the voucherDeliveryType field (possible values: EMAIL_TO_CUSTOMER, EMAIL_TO_AGENT, EMAIL_TO_CUSTOMER_AND_AGENT).

Last but not least, if you have a membership program in place it would be useful to collect the information on when the customer making the booking has joined your membership program. You can send the date when the customer joined using the customerMemberSince field.

All these fields are available under additionalBookingDetails.FraudPreventionDetails in the /bookings/cart/book, allowing the relevant details to be passed along with the booking request and facilitating the fraud prevention process on the Viator side.

Note: If a transaction initiated by a specific user is flagged as potentially fraudulent and you do not include unique identifiers for that user, such as the agentId, our fraud prevention tools will implement heightened security measures across your entire account. This may lead to an increase in failed transactions due to fraud concerns. Conversely, if you provide a unique identifier like the agentId, these measures will only affect future transactions made by that user, resulting in fewer booking failures at the account level. Therefore, we strongly recommend utilizing the fraudPreventionDetails when applicable. Though all these steps will facilitate fraud prevention, this does not guarantee that no transactions will be blocked as fraud prevention, it will only improve the process.

/bookings/cart/book or /bookings/book

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access

Affiliates ✓ Merchants

5. Confirm the booking

- **Confirms/finalizes the held booking** using the /bookings/cart/book or /bookings/book endpoint.
- All **relevant booking information** will need to be provided in the request (e.g. booking questions).
- The bookingRef from the /hold endpoint must be provided in the /book request in order to **confirm the booking from the hold**, otherwise a new booking will be made (with a risk of a price change).
- The **booking status** from the book response must be verified before confirming the booking to the customer.
- **Viator voucher** from the response to the booking endpoint must be shared with customers (custom vouchers are not supported as they won't be recognized by tour operators).

Affiliate partners with booking access must apply **secure voucher redemption**:

- Verify the value returned for the isVoucherRestrictionRequired field in the /bookings/cart/book response for every cart item. This field has value true when the transaction is identified as potentially fraudulent and false when the transaction is deemed legitimate.
- For any potentially fraudulent transactions ("isVoucherRestrictionRequired" : true) take the relevant steps to share the voucher securely **via email** instead of displaying it on the booking confirmation page.

Important: Only the following combination of endpoints is allowed:

- /bookings/cart/hold + /bookings/cart/book
- /bookings/hold + /bookings/book

We highly recommend using the/cart endpoints (/bookings/cart/hold + /bookings/cart/book) as they **support multiple items**.

Affiliate partners with full + booking access are **required** to use the /cart endpoints.

/bookings/status

⊗ **Basic-access Affiliates** ⊗ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

6. Share booking status updates for pending bookings

- Requests the **status** of an existing booking. Statuses can be either: confirmed, rejected, cancelled, pending, in progress, on hold, or failed.
- This endpoint should be used for **periodic checks of the booking status** when the/book response returns a **pending status** (manual confirmation type products) – we recommend **hourly** checks of the booking status. It's important to keep customers up to date on the status of their bookings.
- This endpoint **should not** be used to verify the booking status of all bookings (especially if the response to the booking endpoint hasn't been returned yet). The booking endpoint (/bookings/book or /bookings/cart/book) returns the booking status.
- **Supplier cancellations should not be pulled with this endpoint** but with the /bookings/modified-since endpoint (the process to automate supplier cancellations is described in this article: **Automating supplier cancellations in V.2 Partner API**).
- This endpoint must be called to check the booking status whenever the booking endpoint **returns an error or the response times out**, or for any other reason it's not clear if the booking has been confirmed (it must be called prior to re-booking).
- This endpoint is also useful when reconciling your finances to account for any cancelled bookings where a refund was processed.
- This endpoint is required in order to support **manual confirmation products**.

↑ Back to top

Cancelling bookings

Merchant API partners: All cancellations (except those requested after the date of travel) must be performed via the API, the implementation of the **cancellation API worflow** is **required**. Should you need to cancel a booking after the travel date, please contact Viator Partner Support.

Affiliate API partners with booking access: The cancellation API workflow can be implemented by the partner although it's **not mandatory**. Based on the contractual agreement Viator is the merchant of record and customers can cancel bookings by contacting Viator customer support using the contact details displayed on vouchers.

Learn more about cancellation policies and processes here: **All you need to know about cancellations**.

/bookings/cancel-reasons

⊗ Basic-access Affiliates **⊗ Full-access Affiliates** **✓ Full + Booking access Affiliates** **✓ Merchants**

1. Get cancellation reasons

- In order to successfully cancel a booking, a traveler must select one of our pre-set **cancellation reasons**. This endpoint pulls in the acceptable cancellation reasons. You can take these cancellation reasons and provide them to travelers during the cancellation process.
- **Customer cancellation reasons** can be obtained by providing the value "CUSTOMER" in the type query parameter in the request to this endpoint. When type is not specified, it defaults to "CUSTOMER".
- Recommended: as the acceptable reasons for cancellation may be altered at any point, we recommend retrieving an up-to-date list from this endpoint **monthly**.

/bookings/{booking-reference}/cancel-quote

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

2. Get cancellation quotes

- To see if a booking is eligible to be cancelled, make a call to the API using the **booking reference number** (bookingRef). Note: It's not enough to display the cancellation policy wording as it doesn't guarantee that travelers will understand the amount that will be refunded, i.e. they may not calculate correctly the time left to the start time based on the supplier's time zone. That's why it's important to **double-check the refund amount** using this endpoint to ensure 100% accuracy.
- If the booking can be cancelled you will be returned a **quote** and can proceed with cancellation of the booking using the /bookings/{booking-reference}/cancel endpoint.
- Important: The **"CANCELLABLE" status** doesn't mean that the booking is refundable, make sure to check the **refundAmount** and **refundPercentage** fields.

/bookings/{booking-reference}/cancel

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

3. Cancel a booking

- If the /bookings/{booking-reference}/cancel-quote endpoint identifies the booking as "CANCELLABLE" and the customer accepts the refund amount, you can process the cancellation using /bookings/{booking-reference}/cancel.
- Make sure that the cancellation has been **processed successfully** before communicating it to the customer ("status": "ACCEPTED").

↑ Back to top

Supplier cancellations

Traditionally API partners were relying on emails sent by Viator in order to inform their customers about cancelled bookings. Now you can opt out from the cancellation emails and automate the process for supplier cancellations using API endpoints.

The process to automate supplier cancellations is described in this article: **Automating supplier cancellations in V.2 Partner API**.

/bookings/cancel-reasons

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

1. Map supplier cancellation reasons

- When cancelling a booking, suppliers must select one of our **pre-set cancellation reasons**.
- This endpoint returns all supplier-initiated cancellation reasons when the query parameter **type** is set to "SUPPLIER".
- Retrieve the supplier cancellation reasons provided by Viator and map them to the cancellation reasons available in your system. You can use this information to communicate cancellations to customers.

/bookings/modifed-since

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

2. Get supplier notifications

- This endpoint provides all **booking-event notifications** relevant to the partner (currently only booking cancellations). Currently these event types are supported:

– "CANCELLATION" – the booking was canceled by the **supplier**
– "CUSTOMER_CANCELLATION" – the booking was cancelled by the **customer** (**Note**: This applies only to bookings done by affiliate partners with booking access where Viator acts as a merchant of record and the customer can contact Viator support to request the cancellation. Merchant partners must process customer cancellations on their end and therefore shouldn't expect this event type as cancellations actioned via /bookings/{booking-reference}/cancel endpoint will not be returned. Only if a booking is manually cancelled by the Viator support team as customer cancellation upon merchant partner's request (due to unique circumstances), the cancellation will be returned as a customer cancellation).

- The flow is the same as in case of other /modified-since endpoints – **paginating** with the cursor parameter to pull **all notifications**, with an option to request the data from the point in time indicated in the modified-since parameter when needed (instead of using the cursor parameter).
- The acknowledgeBy timestamp returned in the response indicates by when this event notification must be **acknowledged** using the /bookings/modified-since/acknowledge endpoint.

/bookings/modified-since/acknowledge

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

3. Acknowledge that booking notifications have been received in order to stop email notifications

- Used to **acknowledge receipt of booking event notifications** returned by the /bookings/modified-since endpoint
- When acknowledged, the email notification for the relevant booking **will not be sent**, otherwise an email notification will be sent to the relevant email address (for booking cancellations) provided in the Partner Dashboard.

↑ Back to top

Exchange rates

/exchange-rates

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Get exchange rates

- All our products are priced in the supplier's currency and will need to be converted to the end user. Using the /exchange-rates endpoint you can pull the exchange rates for conversions between specified currencies.
- Exchange rates must be **cached** and refreshed as indicated in the **expiry timestamp**.

Supported currencies

Merchant API partners: While we support many currencies, payments for bookings can only be made using the following four currencies:

- GBP (British Pound)
- EUR (Euros)
- USD (US Dollars)
- AUD (Australian dollars)
- CAD (Canadian dollars)

While you can still price products in any of the supported currencies, you will be invoiced in your choice of the currencies above.

Note: each booking is invoiced in the currency provided in the booking request. If you make bookings in multiple currencies, you will receive separate invoices for each currency.

Affiliate API partners with booking access: The following currencies are supported for bookings: USD, EUR, GBP, AUD, CAD, CHF, DKK, FJD, HKD, JPY, NOK, NZD, SEK, SGD, THB, ZAR, INR, BRL, TWD, MXN, CLP, IDR, ILS, KRW, PHP, PLN, TRY.

You will receive commission payments in the currency selected for payouts in your Partner Dashboard.

↑ Back to top

Traveler reviews

/reviews/product

⊗ **Basic-access Affiliates** ✓ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

Get traveler reviews

- This endpoint retrieves **traveler reviews, supplier responses, review photos** and all other necessary attributes in one request.
- Reviews are retrieved on an **individual product basis**.
- **"Helpful votes"** left by other travelers help provide relevance to customers
- You can retrieve up to 500 reviews per product in one request. Our research indicates that there are diminishing returns after the 100th review. However if you wish to pull all reviews for the product, use the **pagination method** with the start and count parameters.
- You can **filter** reviews by rating, by a single language, and can filter out machine-translated reviews. Machine translated reviews will always be sorted after reviews in the primary language. You can also filter by review **provider** (Viator, Tripadvisor).
- You can **sort** reviews by one of the following: by highest rating, most recent, or most helpful. This sort is applied within the chosen languages.
- Reviews must be **cached** and refreshed **weekly**, as well as **on-demand** when you see that the product content endpoint returns a different review count than saved in your database for the product.
- Review content is **protected proprietary information**; therefore, you may not allow review content to be indexed by search engines. See the details of Viator **non-index policy** here.

Determining ratings

Product ratings and rating counts are returned by the /products/{product-code}, /products/bulk, and products/modified-since endpoints. Details of attributes included can be found in the API specs.

Review display

In case you are displaying Viator or Tripadvisor reviews (or review rating) from the API, please make sure that the following rules are followed:

- all reviews are displayed for a product with the correct review rating instead of selective display of reviews with the highest rating;
- the provider of the reviews (Viator/Tripadvisor) is indicated (for example by including the phrase “collected by Viator and Tripadvisor”)
- all Viator/Tripadvisor reviews are rendered non-indexable by search engines.

↑ Back to top

Traveler photos

/reviews/product

⊗ **Basic-access Affiliates** ✓ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

Get traveler-submitted photos

- Traveler photos are **photos that travelers can upload when submitting their reviews**. These photos are Unlike supplier photos, travelers photos are not ingested through any of the /products endpoints.
- Including traveler photos have their benefits. Traveler photos help capture the experience from a traveler's point of view. We only require suppliers to upload a minimum of 4 photos when creating their listing and traveler photos can augment listings with just the minimum amount of photos.
- Including traveler photos is recommended for an **"Excellent" build** to our API, as we know that including traveler photos generally **improves conversion**.

Note: supplier photos are already ingested through our product ingestion endpoints.

↑ Back to top

Supplier details

/suppliers/search/product-codes

⊗ Basic-access Affiliates ⊗ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Get supplier details

- Product content endpoints return the supplier name and reference. **Full information about the supplier** (name, contact details) can be retrieved with the /suppliers/search/product-codes endpoint **per product** (with a limit of 500 products per request).
- Viator collects and shares contact information about suppliers with type “type”: “BUSINESS”, in case of suppliers with “type”: “INDIVIDUAL” the supplier’s information is limited to name.
- Note: Partners acting as a marketplace – i.e. where there is a transaction occurring on the partner site – should be displaying all of the requisite information for suppliers, including the statement whereby the supplier has certified that they will comply with EU law. On Viator.com – we surface that information during checkout, where the traveller can click to get additional details about the supplier to find their details.
- Supplier details should be **cached**, we recommend a **weekly** refresh of this data.

↑ Back to top

Booking questions

/products/booking-questions

⊗ **Basic-access Affiliates** ⊗ **Full-access Affiliates** ✓ **Full + Booking access Affiliates** ✓ **Merchants**

Get booking questions

- Product Product content endpoints return the booking question ids for questions applicable to an **individual product**. Additional information about **all booking questions available in the API** can be retrieved with the /products/booking-questions endpoint.
- This endpoint returns all relevant data that explains how to **categorize** booking questions and **what type of responses is expected** (based on type, group, maxLength, unit fields), as well as **whether the answer to the question is required or not** (required field with options is MANDATORY, CONDITIONAL and OPTIONAL).
- **Labels** provided in the response for each question can be used for the front-end display to the customer (i.e. “label”: “Traveler height in feet or centimeters (required for safety reasons)”).
- It’s important to differentiate between booking questions that must be asked **per-person** (“group”: “PER_TRAVELER”) and booking questions applicable on the **booking level** (“group”: “PER_BOOKING”). Pickup-related questions, i.e. “PICKUP_POINT” must be answered only once per booking whereas questions such as “WEIGHT” or “DATE_OF_BIRTH” must be answered separately for each individual traveler (multiple input fields are needed instead of collecting answers for all travelers in one input field).
- Booking questions related to arrival and departure details must be **validated correctly to avoid booking failures**. The API documentation includes a table with the logic for conditional booking questions that must be **hardcoded into your implementation** (see Conditional booking questions). Please check this article for a step-by-step guide on how to implement the logic correctly: **Implementing Booking Questions**.

- The response from this endpoint must be **cached**, we recommend refreshing it **monthly**.

↑ Back to top

Attractions

These endpoints make it possible for API partners to create pages dedicated to an attraction – e.g. a page that just promotes tours of the Colosseum. A single attraction could have multiple tour options or multiple tour operators and by building an attraction page, you can present all options to your audience.

These endpoints can also be used to display attractions on search results pages (SRPs) and on destination pages.

/attractions/search

✓ Basic-access Affiliates ✓ Full-access Affiliates ✓ Full + Booking access Affiliates ✓ Merchants

Retrieve all attractions

- This endpoint returns **all attractions within a destination** (facilitates mapping destinations to attractions) and is used to help merchandize products by attraction.
- Attractions can be **sorted** by ALPHABETICAL, DEFAULT & REVIEW_AVG_RATING order.
- **Pagination rules apply** to this endpoint: no more than 30 products can be requested in a single call (“count”: 30). The first request should be done with “start”: 1, the second request must include the value of the start parameter for the next result you wish to see. For example, if the first request was done with “start”: 1 and “count”: 30, the second request will be done with “start”: 31 and “count”: 30, the third one with “start”: 61 and “count”: 30 etc.
- **Attraction details** include: destination mapping, Viator attraction URL (Affiliate partners only), number of products + product codes, reviews, images, introduction, overview, details of admission type (free/not free), opening hours, full address.

- When a product is linked to an attraction the relevant attractionId is returned for a product in the response to the product content endpoint. You can **map products to attractions** using the attractionId extracted from the product content response.
- This is also useful for **navigation** as well as **building out basic attractions pages**.
- With this endpoint you can get aggregated **product and review count/rating** for an attraction.
- Attraction data **must be cached** and refreshed **weekly**.

/attractions/{attractionId}

✓ Basic-access Affiliates **✓ Full-access Affiliates** **✓ Full + Booking access Affiliates** **✓ Merchants**

Get details of a single attraction

- This endpoint is used by partners who can't ingest attraction details with the /attractions/search endpoint but instead need to make real-time calls to get the details of a **single attraction** (using the attractionId extracted from the product content response), or when it's necessary to **get details of a new attraction**.
- **Attraction details** include: destination mapping, Viator attraction URL (Affiliate partners only), number of products + product codes, reviews, images, introduction, overview, details of admission type (free/not free), opening hours, full address.
- Attraction data **must be cached** and refreshed **weekly**.

↑ Back to top

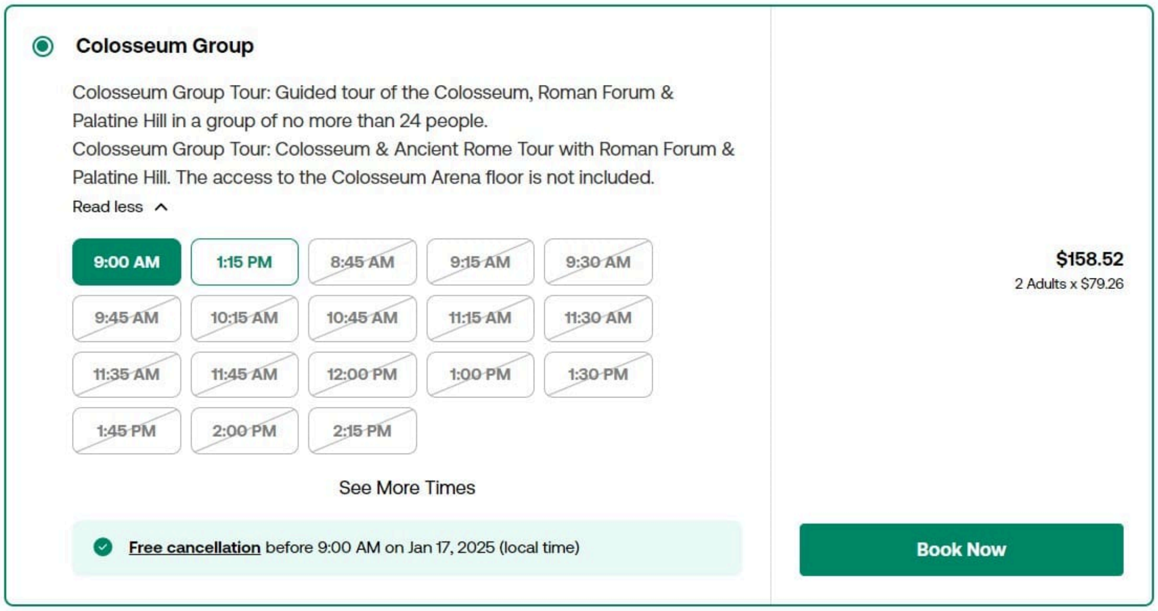
Important validations – product options

Any particular product may consist of a number of variants, each of which is referred to as a ‘product option’. Each product option has an individual setup and there are specific data fields that must be verified correctly on the product option level.

Please refer to our API documentation: Product options.

Product option title and description displayed

Product option title and description must be always displayed to customers for each product option. These fields include essential information about the individual product option and indicate what services are provided for that product option.


Colosseum Group

Colosseum Group Tour: Guided tour of the Colosseum, Roman Forum & Palatine Hill in a group of no more than 24 people.
Colosseum Group Tour: Colosseum & Ancient Rome Tour with Roman Forum & Palatine Hill. The access to the Colosseum Arena floor is not included.

Free cancellation before 9:00 AM on Jan 17, 2025 (local time)

$158.52
2 Adults x $79.26

Language guides validated

Language guides must be validated on the product option level. Suppliers set up language guides on the product option level and they are returned separately for each product option in the API response.

See a sample response for product 16168P10:

Example response for product 16168P10 with different language guides for different options

Each product option can have different language guides and it's important to display this information to customers for each product option based on productOptions.languageGuides in order for them to select one. The guide type must be always displayed as well (audio, written, guide).

Pickup / drop-off verified for product options

Pickup / drop-off must be validated on the product option level. A product could have two different product options – one product option with pickup and another product option without pickup. The front-end and the back-end implementation in both cases would be different.

When the value of logistics.travelerPickup.pickupOptionType in the product content response is "PICKUP_AND_MEET_AT_START_POINT", it means that the product includes both 'hotel pickup' and 'meet at the departure point' variants in its product options; e.g., one product option may be for hotel pickup while another is for meeting at the start/departure point.

To determine which product options offer what, **it is necessary to inspect each product option's details** in the productOptions[] array in the product content response.

- If **pickup is included** for a product option, the **phrase "Pickup included"** will be present in the productOptions[].description field. When booking a product option with pickup included **you must collect a pickup point** from the user that corresponds to one of the entries in the logistics.travelerPickup.locations[] array in the product content response.

Front-end: All booking questions related to arrival/departure must be displayed as normal to allow customers to select the desired arrival/departure mode and provide all relevant information to the supplier before travel.

Back-end: All answers provided by the customer must be captured and submitted in the booking request following the logic for conditional booking questions from the table in the API documentation.

Example product content response for product option with pickup

Example booking request for product option with pickup

Booking response fo the request above

- If the productOptions[].description field **does not contain the phrase "Pickup included"**, this indicates that this product option **does not include pickup** and should be considered to follow a "MEET_AT_DEPARTURE_POINT" arrangement.

Front-end: In case of product options without pickup, you shouldn't ask arrival or departure-related questions at checkout (but you will still need to answer the questions returned in the API using values hardcoded into your implementation). If you wish, you can display at checkout the meeting point location(s) for informational purposes (not collecting answers) but not the pickup locations and no questions related to arrival/departure should be displayed in that case. The option to contact the supplier later (CONTACT_SUPPLIER_LATER) doesn't apply to this scenario either and should not be present as it applies only to product options with pickup.

Back-end: In the answer to the PICKUP_POINT question, you will need to send the location reference MEET_AT_DEPARTURE_POINT (returned under travelerPickup.locations). In case the product returns the TRANSFER_ARRIVAL_MODE or TRANSFER_DEPARTURE_MODE questions, these must be answered with OTHER.

Example product content response for product option without pickup

Example booking request for product option without pickup

Booking response for the request above

You can use these instant confirmation type products for testing: 9025P51, 62450P1.

This article includes a step-by-step guide on how to validate pickup and apply conditional booking questions correctly: Implementing Booking Questions.

Finding meeting points

Meeting point locations may be specified by the supplier on the product option level. When that's the case, the product option description will include information about the meeting point applicable to each product option.

For example, product 131843P24 has two meeting point locations under logistics.start:

Example response for product 131843P24 with multiple meeting points

However they don't apply to all product options. Instead, each product option comes with a different meeting point as specified in the productOption.description:

Response for product 131843P24 with product option descriptions

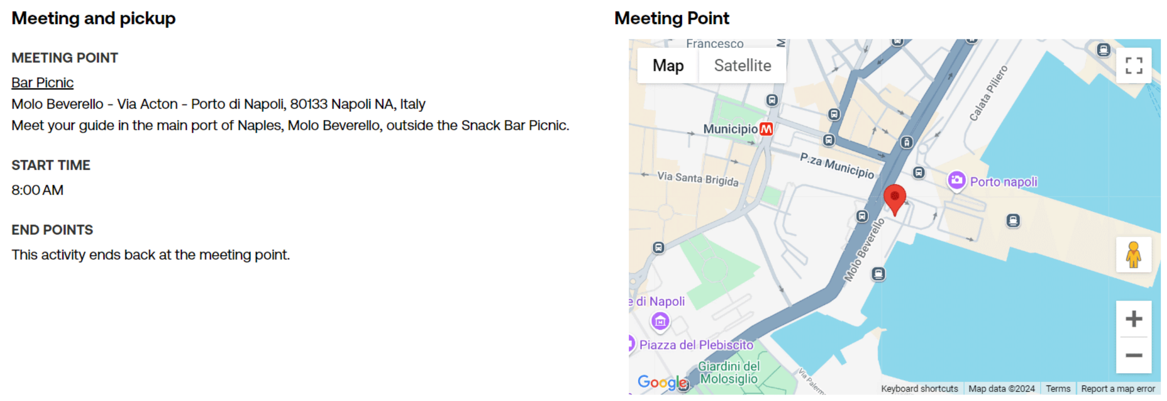
When booking a product/product option with the meeting point, **you won't be asked** to provide the meeting point location selected by the customer in the booking request.

However **it's recommended to display the meeting point locations on product pages** so that customers are aware of the available options. Most importantly, you will need to make sure that the product option description returned in the API response is clearly visible as it provides guidelines about the meeting point for each product option.

When a product returns the PICKUP_POINT question, in order to book the product option without pickup, or allow the customer to meet at the meeting point instead of selecting the pickup location (for a product option with pickup), you will need to answer the PICKUP_POINT question with a special location reference: MEET_AT_DEPARTURE_POINT.

In case customers contact you to confirm the meeting point location post booking, you can refer to the **meeting point location displayed on their Viator voucher**.

In case of product 131843P24, the voucher for a booking made for product option TG1 ("CAPRI FROM NAPLES") will include the meeting point location set up by the supplier for the selected product option:



Meeting and pickup

MEETING POINT

Bar Picnic
Molo Beverello - Via Acton - Porto di Napoli, 80133 Napoli NA, Italy
Meet your guide in the main port of Naples, Molo Beverello, outside the Snack Bar Picnic.

START TIME

8:00 AM

END POINTS

This activity ends back at the meeting point.

Meeting Point

Adherence to endpoint usage rules

To ensure that the load on our systems is not excessive, we ask that you develop your API implementation in accordance with our endpoint-specific usage rules. If you have a use case which would violate a rule, you must first seek approval from our API Onboarding Team.

Please refer to our API documentation: Update frequency.

Fixed-cadence delta updates

You must adhere to the following update frequency guidelines. More frequent updates will place an excessive burden on our systems and may result in your integration being shut off.

See section on Rate Limiting for more information.

Fixed cadences for polling content ingestion endpoints

Fixed-cadence full updates

To ensure your systems reflect any removals of or changes to existing auxiliary data, we require you to retrieve full updates from these endpoints as follows:

Fixed cadences for polling content ingestion endpoints

On-demand updates

When ingesting product content, if you encounter an unknown reference (e.g., a new location reference, booking question, tag, or destination ID), or if you need to perform a currency conversion and the last exchange rate you retrieved has expired, or if any other condition outlined below is met that requires retrieving additional data, you must call the relevant endpoint to resolve the new reference immediately or just after completing the product content update.

Fixed cadences for polling content ingestion endpoints

↑ Back to top

Additional resources

Make sure to check:

1. API documentation

- Access to endpoints
- Update frequency
- Product content and availability endpoints

2. Additional guides from our Partner Resource Center

- Managing product and availability data
- Front-end guide
- Viator API Certification
- Viator API certification: back-end checks

Implementation Guides

Technical Guide

Front-End Guide

Viator Tags

Glossary

Implementing Age Bands

Cancellation Policies

Viator API Certification

Viator API Certification – back-end checks

Implementing Traveler Photos

Golden Path | Basic Access Affiliate Partners

Affiliate Attribution: How It Works

Managing Product & Availability Data

Calculating Product Pricing

Implementing Booking Questions

API Amendments Workflow

Automating Supplier Cancellations

Booking reporting guide

Help center

Privacy & Cookies Statement

About Viator

Terms & Conditions