



Documento Integración API

Sistema Venta Turismo

Versión: v1.15 Externos

Fecha: 25-10-2024

Bmore

La información contenida en este documento está catalogada como reservada y de uso interno en **Bmore**.
Su divulgación y/o reproducción total o parcial requiere autorización expresa y escrita por parte de **Bmore**.

CONTROL DE CAMBIOS DEL DPS

VERSIÓN	FECHA	CAMBIOS RESPECTO DE LA VERSIÓN ANTERIOR	PREPARADO POR	REVISADO POR	APROBADO POR
0.9	15-02-2023	Borrador	Rodolfo Contardo		
1.0	16-02-2023	Inicial. Se agregan diagramas de interacción, se cambia addTime por resetTime, se completa Response de Rater, se cambia redacción de varios textos. Se especifica mejor algunos campos	Rodolfo Contardo		
1.1	17-02-2023	Se quita SaleChannel en todos lados Se quita todos los valores afectos (taxable) y exentos (Exempt) Se cambia parámetros de entrada y salida a availabilityInformation y se cambia de nombre a tourInformation En rater se quita en request Items de detalle y se mejora diagrama de interacción En AddToCart se quita WaitTime en request Se mejora descripción en los pasos en punto 2.7 Booking y Pago CouponCode se quita de API Pay y se agrega a Booking 2.8 se modifica diagrama de interacción	Rodolfo Contardo		
1.2	23-02-2023	- Se modifica flujo Booking y Pago No Exitoso agregando flujo alternativo para limpiar booking - Se cambia POST /api/v1/Booking a /api/v1/startPayment - /api/v1/removeFromCart se agrega respuesta del tipo de eliminación (P o T) - /api/v1/tourInformation se agrega parámetro opcional NumberDays y se agrega respuesta con ejemplo quiebre por cambio tarifa	Rodolfo Contardo		
1.3		Se modifica Llamado a /api/v1/removeFromCart parámetros vayan en la URL	Rodolfo Contardo		
1.4	30-05-2023	Se cambia especificación de API TourCodes. Se agrega información de llamado a token y URL de ambientes de prueba.	Rodolfo Contardo		
1.5	15-06-2023	- Se detalla la ejecución en POS de /connect/token - Se agrega en rater, startPayment y pay el anticipo de pago (campos AdvancePaymentAmount AdvancePaymentPercentage HasAdvancePayment)	Gonzalo Pino y Rodolfo Contardo		
1.6	27-06-2023	- Se reconfigura request de AddtoCart para que reciba los parámetros de entrada, y la salida se agrega BookingExpirationDate y se cambia tipo de datos para WaitTime - TourInformation al response se agrega startTime y una colección para los PickupLocations de la excursión consultada.	Rodolfo Contardo		

1.7	27-07-2023	Se modifica response para resetTime. Se agrega los intervalos para los servicios, particularmente los relacionados al Hop On, se cambia el nombre a la estructura de Quotas, se agrega Pickup location y meeting points en tourInformation.	Rodolfo Contardo		
1.8	31-01-2023	Se agrega ejemplos de referencia json de para request y response	Rodolfo Contardo		
1.9	02-08-2023	Se agrega MeetingPointId, PickupLocationId y StartTime a AddToCart Mejora en la descripción de validaciones en AddtoCart. Se especifica las consideraciones en los elementos que son opcionales al conformar json (texto introductorio Capitulo 3)	Rodolfo Contardo		
1.10	11-08-2023	Se agrega descripción para endpoint AddPassenger y Pay. Se modifica respuesta de DELETE /api/v1/booking. Se agrega descripción de validaciones adicionales a addtoCart, removeFromCart y resetTime. Se mejora descripción e ilustración de Agregar a Carrito y Agregar Nueva Excursión a Carrito	Rodolfo Contardo		
1.11	25-08-2023	Se Cambia Response en /api/v1/tourcodes para reflejar las divisas disponibles en OzyTrip. En Request /api/v1/tourInformation campo opcional Currency. En Request /api/v1/rater campo opcional Currency y en Request api/v1/pay campo opcional Currency. Mejora en descripción puntos 2.5. Endpoint /api/v1/tourInformation NumberDays se cambia a requerido	Rodolfo Contardo		
1.12	14-11-2023	Se agrega en response Rater el identificador del Item eCommerce (idItemEcommerce) Se agrega URL de ambiente productivo	Rodolfo Contardo		
1.13	26-03-2024	Se agrega AnonymousPassenger en addPassengers con la posibilidad de dejar una lista vacía. Afecta en si se debe llenar o no la lista de Passengers e ItemsCart. En Rater se agrega en forma global el item DiscountComment en la respuesta	Rodolfo Contardo		
1.14	12-08-2024	Se incorpora respuesta JSON en caso de llamar nuevamente por idBooking que ya fue pagado, en vez de Bad Request. Se especifica en punto 3.10.3.	Rodolfo Contardo		
1.15	25-10-2024	Se incorpora nueva información de conexión ambiente de desarrollo para integraciones externas	Rodolfo Contardo		

Tabla de Contenidos

1	INTRODUCCIÓN	6
2	ESCENARIOS DE INTERACCIÓN	7
2.1	CONSULTA DISPONIBILIDAD	7
2.2	AGREGAR A CARRITO	8
2.3	AGREGAR NUEVA EXCURSIÓN A CARRITO	9
2.4	QUITAR EXCURSIÓN DE CARRITO	10
2.5	CÁLCULO TARIFAS	11
2.6	BOOKING Y PAGO	12
2.7	BOOKING Y PAGO NO EXITOSO	14
3	API INTEGRACIÓN	16
3.1	POST /CONNECT/TOKEN	17
3.1.1	Request.....	17
3.1.2	Response	18
3.2	GET /API/V1/TOURCODES	19
3.2.1	Request.....	19
3.2.2	Response	19
3.3	GET /API/V1/TOURINFORMATION	20
3.3.1	Request.....	20
3.3.2	Response	21
3.4	POST /API/V1/RATER.....	24
3.4.1	Request.....	24
3.4.2	Response	24
3.5	POST /API/V1/ADDCART	26
3.5.1	Request.....	26
3.5.2	Response	27
3.5.3	Validaciones	27
3.6	DELETE /API/V1/REMOVEFROMCART	29
3.6.1	Request.....	29
3.6.2	Response	29
3.6.3	Validaciones	29
3.7	PUT /API/V1/RESETTIME	30
3.7.1	Request.....	30
3.7.2	Response	30
3.7.3	Validaciones	30
3.8	DELETE /API/V1/BOOKING	31
3.8.1	Request.....	31
3.8.2	Response	31
3.8.3	Validaciones	31
3.9	POST /API/V1/ADDPASSENGERS.....	32
3.9.1	Request.....	32
3.9.2	Response	34
3.10	POST /API/V2/ADDPASSENGERS	35
3.10.1	Request	35
3.10.2	Response	37
3.11	POST /API/V1/PAY	38
3.11.1	Request	38
3.11.2	Response	39

3.11.3	Validaciones	39
--------	--------------------	----

1 Introducción

En este documento se presenta las interacciones habituales entre un eCommerce y OzyTrip eCommerce API donde se mostrará una serie de diagramas de interacción, y finalmente, el detalle de cada API ocupada en estos diagramas en el capítulo 3 API Integración, con el fin de facilitar el entendimiento de cómo utilizar OzyTrip eCommerce API.

2 Escenarios de Interacción

En este capítulo se muestra escenarios habituales de integración para un eCommerce estándar, donde se podrá ver consulta de tarifas, agregar a carrito, realizar el booking y posterior registro del pago entre otros.

2.1 Consulta Disponibilidad

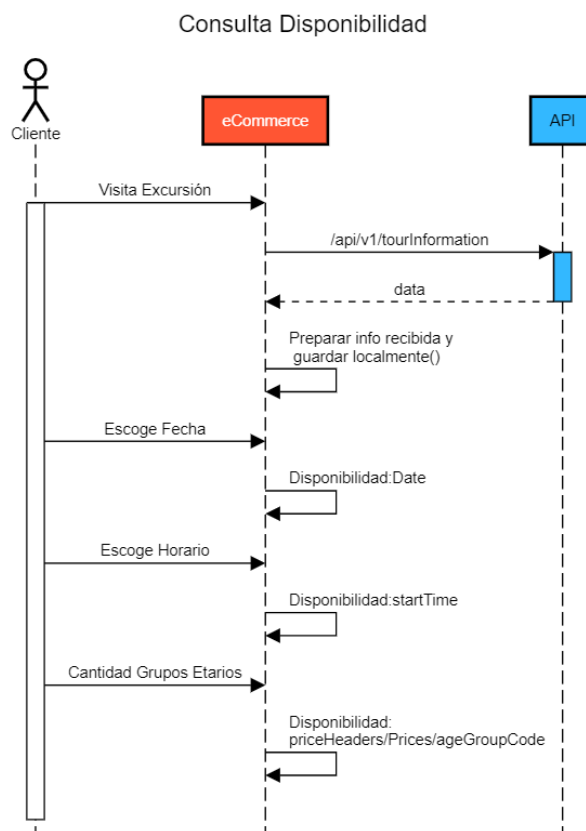


Ilustración 1 - Consulta disponibilidad

En este escenario cliente selecciona la excursión de su agrado, donde el eCommerce podrá responder con la información necesaria al consultar `/api/v1/tourInformation`. Éste le proveerá de datos de disponibilidad de cupos por fecha, precios por grupo etario, horarios, puntos de encuentro con la hora en que el pasajero se puede presentar en dicho lugar, todo esto realizando una sola consulta.

2.2 Agregar a Carrito

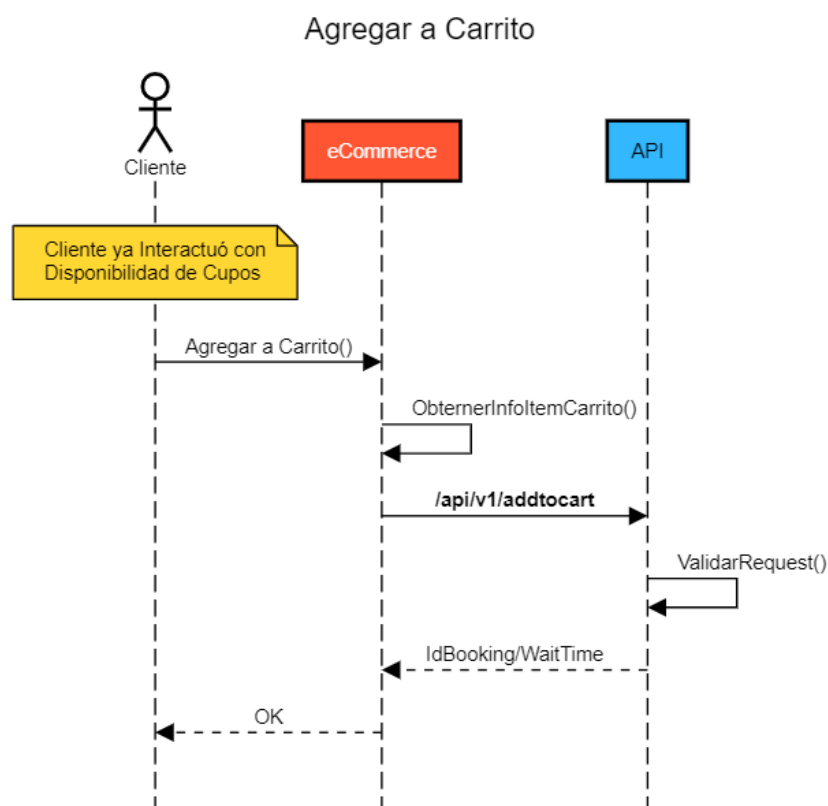


Ilustración 2 - Agregar a Carrito

Cliente ya tiene claro que desea comprar por tanto presiona botón Agregar a carrito o Reservar ahora. Si no hay ningún elemento en el carrito previamente, el campo `IdBooking` debe ir en null.

Esta API realizará una serie de validaciones, entre ellas de completitud de campos, de cupos para la excursión señalada, fechas y cantidad de pasajeros por grupo etario (en este momento aún no es necesario entregar información detallada de cada pasajero).

Si todo va bien se devolverá `IdBooking`, `WaitTime` que es la espera en carrito antes que éste sea borrado automáticamente y `bookingExpirationDate` que indica la hora exacta en que el carrito será borrado.

Donde el flujo sería el siguiente:

1. Cliente agrega un ítem al carrito de eCommerce (`Agregar a Carrito()`)
2. Se obtiene para dicho ítem (`IdItemEcommerce`) lo que se tiene actualmente en carrito y con esta información se llama a EndPoint `/api/v1/addtocart` (`ObtenerInfoItemCarrito()`). Esto para evitar diferencia en lo que se tiene en carrito eCommerce vs el carrito en OzyTrip
3. Se realiza validaciones (`ValidarResultado()`).
4. Si va todo bien se arma respuesta.

2.3 Agregar Nueva Excursión a Carrito

Agregar Nueva Excursión a Carrito

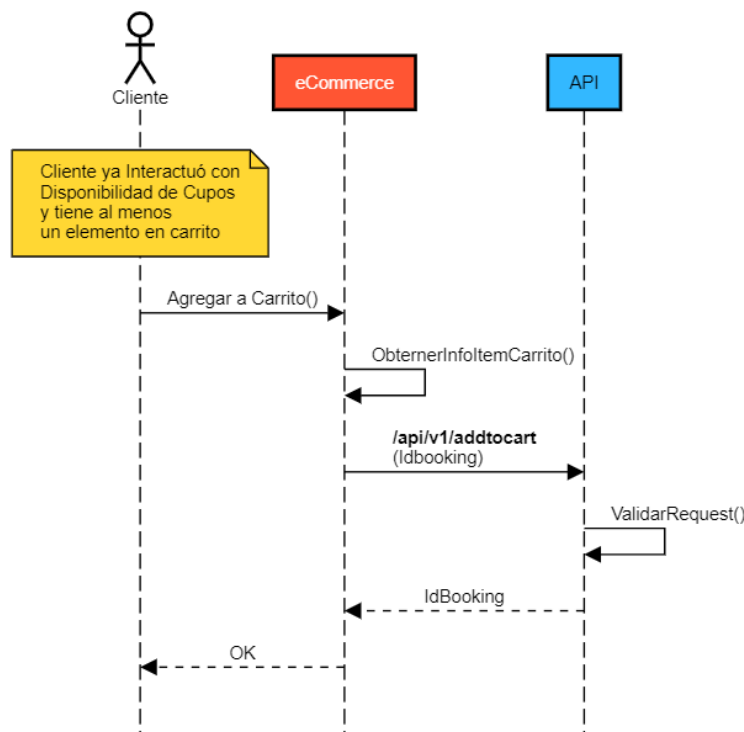


Ilustración 3 - Agregar a carrito una nueva excursión

El escenario es el mismo que el punto 2.2 Agregar a Carrito, pero además ya tiene uno o más elementos en el carrito. En este caso eCommerce debe enviar la información indicando el mismo `IdBooking` que se le entregó previamente cuando agregó por primera vez una excursión al carrito.

2.4 Quitar Excursión de Carrito

Quitar Excursión a Carrito

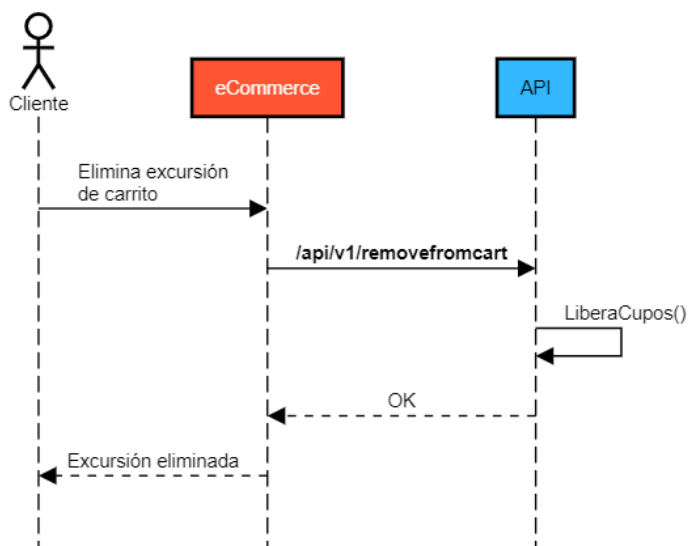


Ilustración 4 - Quitar una excursión del carrito

Cliente al estar en carrito elimina una excursión de éste, para ello eCommerce debe llamar a `/api/v1/removeFromCart` indicando `IdBooking` y `IdItemEcommerce` donde se liberarán los cupos asociados a esta excursión.

Nota Importante: Si al eliminar la excursión, resulta que es la última, el `IdBooking` queda sin efecto donde no podrá ser utilizado nuevamente. Si se llegara a enviar a `/api/v1/addToCart` este mismo `IdBooking` se devolverá un error.

2.5 Cálculo Tarifas

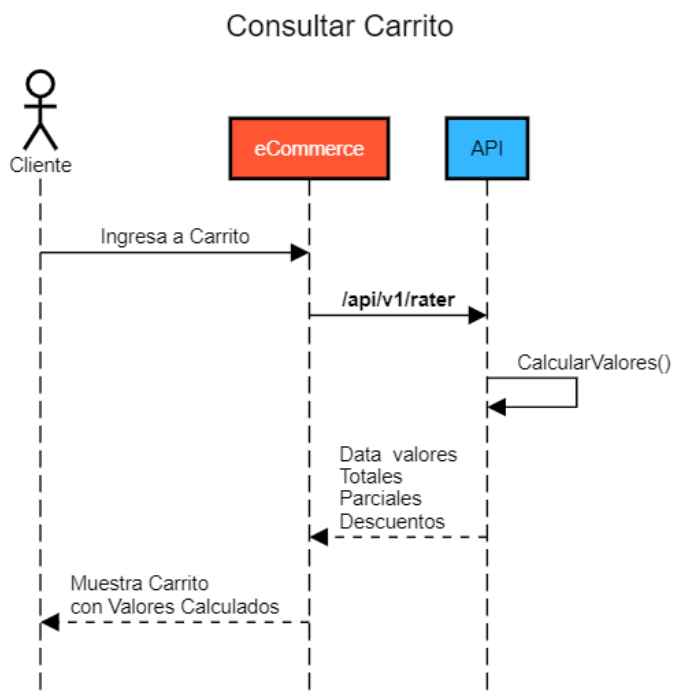


Ilustración 5 - Consultar tarifas al revisar carrito

Cliente al revisar el carrito, eCommerce podrá consultar la tarifa de todo lo que se encuentre en ese momento en el carrito utilizando la `/api/v1/rater`, para ello debe indicar:

- `IdBooking`
- Un cupón (este dato es opcional)
- Medio de Pago (este dato es opcional)
- Currency (este dato es opcional, de no indicar se ocupará CLP. De indicar solo debe ser un valor del listado entregado por endpoint `/api/v1/tourcodes`)

La respuesta contendrá todos los valores necesarios para mostrar al cliente por cada excursión y fecha del servicio, esto es:

- Valores totales.
- Pago anticipado y el porcentaje que corresponde este pago anticipado
- Valores parciales por grupo etario.
- Descuentos si aplicara.

2.6 Booking y Pago

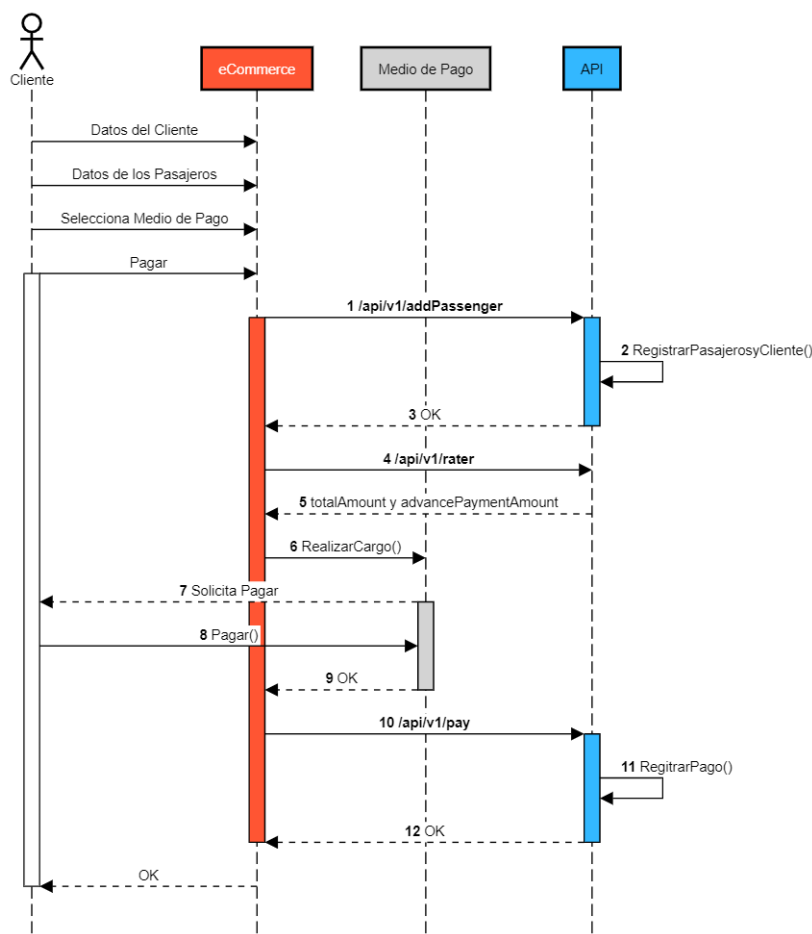


Ilustración 6 - Booking y pago

Cliente debe ingresar los datos personales de contacto, lo anterior para el envío y/o notificación de la información de venta y documentos de viaje. Adicionalmente le serán requeridos los datos de los pasajeros que lo acompañan, para todos los servicios que tenga en el carrito; para luego seleccionar el medio de pago de su preferencia. Una vez realizado esto presionará botón Pagar.

Flujo de la ilustración 6 - Booking y pago:

1. eCommerce llama a `/api/v1/addPassenger` ingresando los datos de contacto, pasajeros con sus grupos etarios y datos personales.
2. API registra todos los datos entregados, además de validar la completitud de estos particularmente para el caso de las excursiones que solicitan todos los datos de contacto de los pasajeros, esto es dependiendo si son `NONE`, `REGULAR` o `MANDATORY`. Esta información se obtiene desde `/api/v1/tourInformation`, Adicionalmente, da un tiempo adicional extendido para que se pague este carrito.
3. API da OK a eCommerce para continuar con el siguiente paso.

4. Se llama a `/api/v1/rater` para obtener `totalAmount` y `advancePaymentAmount` a ocupar dependiendo si se realiza pago completo o anticipo. Se debe tener especial cuidado en utilizar la misma divisa
5. Recibe información de API Rater.
6. eCommerce instruye al Medio de Pago para que realice el cargo.
7. Medio de Pago interactúa con Cliente para pagar.
8. Cliente intenta realizar el Pago.
9. Medio de Pago da OK con el cargo.
10. eCommerce llama a `/api/v1/pay` ingresando todos los datos del pago, así como el Nro de Orden o Nro de Pedido.
11. API registrará los datos del pago.
12. eCommerce informa al cliente que todo resultó ok.

2.7 Booking y Pago No Exitoso

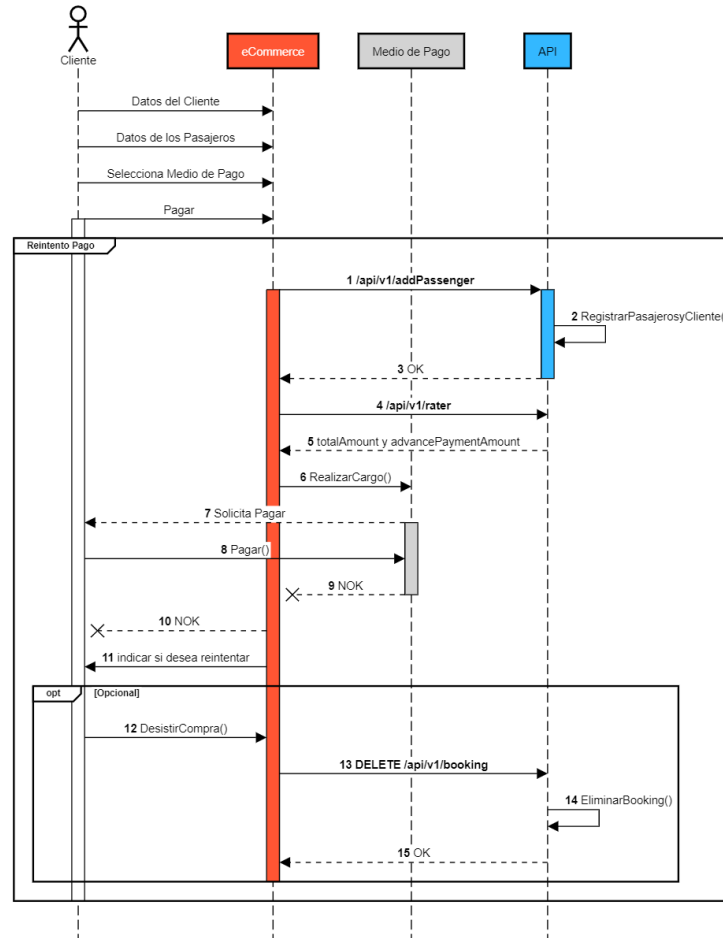


Ilustración 7 - Booking y pago no fue exitoso

Cliente debe ingresar los datos personales de contacto, lo anterior para el envío y/o notificación de la información de venta y documentos de viaje. Adicionalmente le serán requeridos los datos de los pasajeros que lo acompañan, para todos los servicios que tenga en el carrito; para luego seleccionar el medio de pago de su preferencia. Una vez realizado esto presionará botón Pagar.

Flujo de la ilustración 7 - Booking y pago no fue exitoso:

1. eCommerce llama a `/api/v1/addPassenger` ingresando los datos del cliente, pasajeros con sus grupos etarios y datos personales.
2. API registra todos los datos entregados, además de validar la completitud de estos particularmente para el caso de las excursiones que solicitan todos los datos de contacto de los pasajeros, esto es dependiendo si son `NONE`, `REGULAR` o `MANDATORY`. Esta información se obtiene desde `/api/v1/tourInformation`, Adicionalmente, da un tiempo adicional extendido para que se pague este carrito.
3. API da OK a eCommerce para continuar con el siguiente paso.

4. Se llama a `/api/v1/rater` para obtener `totalAmount` y `advancePaymentAmount` a ocupar dependiendo si se realiza pago completo o anticipo
5. Recibe información de API Rater.
6. eCommerce instruye al Medio de Pago para que haga el cargo.
7. Medio de Pago interactúa con Cliente para pagar.
8. Cliente intenta realizar el Pago.
9. Medio de Pago indica rechazo al cargo (NOK) ya sea por cancelación, por error en tarjeta u otro motivo, e informa a eCommerce.
10. eCommerce presenta al cliente falla con dicho medio de pago al intentar pagar.
11. eCommerce le presenta posibilidad de reintentar pago, ya sea con el mismo medio de pago u otro, si tuviera configurado uno adicional. **Si en este nuevo intento el Cliente logró pagar, es decir, Medio de Pago dice OK, el flujo restante es el mismo al señalado en Capítulo 2.7 Booking y Pago, punto 8 en adelante.**

Flujo Opcional (con el fin de no dejar eternamente los cupos reservados se puede ocupar este flujo opcional. Aunque en back existirá un Cron que hará limpieza de estos casos pasado la fecha de expiración de la reserva, siempre y cuando ésta aún no esté en un estado terminal)

12. Se desiste de la compra.
13. eCommerce realiza llamado a API DELETE `/api/v1/booking`.
14. Back realiza eliminación de booking de forma lógica.
15. API devuelve OK de la acción.

3 API integración

En este capítulo se detalla todas las API de integración, indicando como se debe realizar el llamado (request) y que datos responderá (response) además de indicar los tipos de datos, su obligatoriedad, ejemplos y una pequeña descripción.

Cabe señalar que todas las llamadas deben ser realizadas con un token válido el cual se detalla en el punto 3.1 con la información señalada en la tabla request parámetros.

Se debe utilizar la lista entregada por endpoint `/api/v1/tourCodes` en los endpoint que soliciten dicha información para sus consultas, por ejemplo: `/api/v1/tourInformation`

Los ambientes de prueba:

Token: <https://app01.dev.bmore.cl:44316/connect/token>

API: <https://api.dev.bmore.cl:8443/>

API Swagger: <https://api.dev.bmore.cl:8443/swagger/index.html>

Para producción:

Token: <https://ozyidentity.bmore.cl/connect/token>

API: <https://ozytripapi.bmore.cl>

Como norma general los elementos que son indicados como opcionales, o están bajo una condición que no cumplen, estos siempre deben ser agregados al request con valor nulo (null), por ejemplo:

```
{
  "idBooking": null,
  "tourCode": "55e6b389-4238-4935-9268-a48390ke54rb",
  "serviceDate": "2023-08-05T12:00:00",
  "startTime": "12:00:00",
  "meetingPointId": null,
  "pickupLocationId": null,
  "ageGroups": [
    {
      "idItemEcommerce": "e9c02524b83644fdb0e807f50b536688",
      "ageGroupCode": "ADT",
      "quantity": 2
    },
    {
      "idItemEcommerce": "77e0d4b8d98146968dfe35979e677a47",
      "ageGroupCode": "NIN",
      "quantity": 1
    }
  ]
}
```


3.1 POST /connect/token

Obtiene un token válido para realizar todas las consultas a la Api.

3.1.1 Request

URI

POST /connect/token

Parámetros

Campo	Tipo Dato	Observación	Información adicional
grant_type	String	Ingresar: client_credentials	Campo requerido
client_id	String	Ingresar: EcommerceTuristik2	Campo requerido
client_secret	String	Ingresar: a5d4ca55-20bc-7f62-8cc1-c3721628e898	Campo requerido
scope	String	Ingresar: ozy_trip_ecommerce_api	Campo requerido

La forma de autenticación usada es Basic enviada en la cabecera de la solicitud, de la siguiente forma:

```
POST /connect/token HTTP/1.1
Authorization: Basic <Base64 de client_id:client_secret>
```

Los datos que deben ir en el body solamente son: grant_type y scope y el content-type debe ser application/x-www-form-urlencoded,

Ejemplo Request

```
POST /connect/token HTTP/1.1
Accept: application/json
Authorization: Basic RWNvbWllcmNlVHVyaXN0aWs6NmVkZWRIZmMtMTUyYS1hYzdjLWY0MDYtYWVjMjQ2ODIyOWE0
Content-Length: 58
Content-Type: application/x-www-form-urlencoded
Host: 192.168.0.10:15892

grant-type: client-credentials
scope: ozy_trip_ecommerce_api
```

3.1.2 Response

Campo	Tipo Dato	Observación	Información adicional
access_token	String	Token a ocupar	
expires_in	Numero	Tiempo de expiración	
token_type	String	Tipo de Token "Bearer"	
scope	String	Scope utilizado	

Ejemplo Response

```
{
  "access_token":
  "eyJhbGciOiJSUzI1NiIsImtpZCI6IjkwNEQzMtK1NzI2MkM0QzA4NzFBMTFEMTc0OEY3RDdEiwiwidHlwIjoiYXQrand0In0.eyJpc3MiOiJodHRwOi8vMTAuMTAuMS4yNToxMTYxNyIdIm5iZiI6MTY4NTU3MjIwNzIwIiwiaWF0IjoxNjg1NjcyMjA0LCJleHAiOiJ0e2ODU1NzU4MDQsImF1ZCI6Im96eV90cm1wX2Vjb21tZXJjZV9hcGkiLCJzY29wZSI6WyJvenlfdHJpcF91Y29tbWVyY2VfYXBP10sImNsaWVudF9p2CI6IktVjb21tZXJjZVR1cm1zdGlrIn0.dH2RUuVNdBvrzZxpRiJ6Sile0bVcIxUTAAQGCasFiYjO-tmctAnvTcmnsFe8BzRijG1tYEMafyO5QDKG8-CdEYQCdW5uwFe_7qQ6xeB6ERFs0v8qUvQuOiOMRpbY9Z9e3Taesp-3EsqiVKbHcWsZ1R8y67cawxKb0VwpB1r-EYTCfUPcLNKP6c-BbdOufhIz8GptGyjdQ-g_9ReoNErF4TqQGQGVW4F4fP9hhu8Vxs2YtrjKeZVPoWm5sJZZFUA7houEbZQ01a9F2-qCrGbVjen1IrScS2rTzPVDLy7CV7xh8Eaadxqfmp14u_KGG4NVAscDIH2G7EIEVwc2_BxLA",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "ozy_trip_ecommerce_api"
}
```

3.2 GET /api/v1/tourCodes

Obtiene listado de todas las excursiones activas, este listado sirve como consulta de apoyo para las excursiones habilitadas y sus grupos etarios soportados.

3.2.1 Request

URI

GET /api/v1/tourCodes

Sin parámetros

3.2.2 Response

Campo	Tipo Dato	Observación	Información adicional
TourCodes	Collection (T)	Listado de excursiones	
T/TourCode	String	Código de la excursión	
T/SourceCode	String	Origen de la excursión	
T/Name	String	Nombre de la excursión	
T/InformationRequired	String	Si los datos de contacto son necesarios según lo señalado en <i>InformationRequiredDescription</i> REGULAR: Información básica de pasajeros y detalle del cliente MANDATORY: Información detallada de pasajeros y cliente NONE: sin información de los pasajeros, sólo información de Cliente	
T/InformationRequiredDescription	String	Descripción del campo <i>InformationRequired</i>	
T/StartTime	HH:mi:ss	Hora de inicio de la excursión	
T/AgeGroups	Collection	Listado de grupos etarios por cada excursión	
AvailableCurrency	Collection (AC)	Listado de divisas disponibles soportadas por OzyTrip, por ejemplo, CLP, USD, BRL, etc	
AC/Currency	String	Listado de divisas disponibles en OzyTrip según la definición de Códigos de divisa ISO 4217	
AC/Description	String	Descripción de la divisa	

Ejemplo Response

```
{
  "tourCodes": [
    {
      "tourCode": "a42b1bd8-a2d6-40c6-b9e0-b1bcf2251817",
      "sourceCode": "OzyTrip",

```

```

    "name": "Viña Concha y Toro Marques AM",
    "informationRequired": "REGULAR",
    "informationRequiredDescription": "Información básica de pasajeros y detalle del cliente",
    "startTime": "08:45:00",
    "ageGroups": [
      "ADT"
    ]
  },
  ],
  "availableCurrencies": [
    {
      "currency": "CLP",
      "description": "Chilean peso"
    }
  ]
}

```

3.3 GET /api/v1/tourInformation

Ver disponibilidad y cotizar una excursión. Esta API puede ser utilizado en página principal de la excursión, pudiendo realizar una sola consulta al momento de ingresar a dicha página, para luego presentar la información al cliente a medida que va indicando fecha, luego horario y por último los grupos etarios y sus valores unitarios.

3.3.1 Request

URI

`GET /api/v1/tourInformation/{tourCode}/{date}/{numberDays}/{currency}`

Parámetros

Campo	Tipo Dato	Observación	Información adicional
{TourCode}	String	Código de la excursión	Campo requerido
{Date}	yyyy-mm-dd	Fecha de Consulta (usualmente hoy)	Campo requerido
{NumberDays}	numero	Cantidad de días a consultar basado en {date}, es decir devolverá información de tanto días como se indique desde la fecha ingresada en {date}. Este campo no podrá ser superior a 150 días.	Campo requerido
{Currency}	String	Divisa según listado entregado por endpoint /api/v1/tourcodes	Opcional (De no indicar se ocupará CLP)

Ejemplo de llamado:

```
GET http://server:port/api/v1/tourInformation/d77a1159-09db-49a6-bafe-6936bafdc65/2023-02-01/15
```

3.3.2 Response

Campo	Tipo Dato	Observación	Adicional
TourCode	String	Código de la excursión	
TourName	String	Nombre de la excursión	
StartTime	HH:mi:ss	Hora de inicio de la excursión	
InformationRequired	String	Si los datos de contacto son necesarios según lo señalado en <code>InformationRequiredDescription</code> REGULAR: Información básica de pasajeros y detalle del cliente MANDATORY: Información detallada de pasajeros y cliente NONE: Solo información de Cliente	
InformationRequiredDescription	String	Descripción del campo <code>InformationRequired</code>	
EncounterType	String	Define el tipo de encuentro si el servicio utiliza Pickup Locations o Meeting Points, donde puede ser: NONE: No utiliza Pickup ni Meeting PICKUPLOCATION: Solo utiliza listado de <code>PickupLocations</code> MEETINGPOINT: Solo utiliza listado de <code>MeetingPoints</code>	
EncounterTypeDescription	String	Descripción del tipo de encuentro	
PriceHeaders	Collection (PH)	Precios según rango de fecha (<code>InitDate</code> y <code>EndDate</code>). Podrá venir más de un rango de fecha dependiendo de cambios de precio	
UseIntervals	Boolean	Indica si se debe ocupar o no <code>SalesInterval</code> Esto en particular para excursiones como Hop on Hop Off	
PH/InitDate	yyyy-MM-dd	Fecha inicio de los precios, junto a <code>EndDate</code> señala un rango de tiempo para consultar el precio de la excursión.	
PH/EndDate	yyyy-MM-dd	Fecha final de los precios	
PH/Prices	Collection (P)	Para <code>UnitPrice</code> se debe realizar la búsqueda por <code>AgeGroupCode</code> , <code>DayType</code> y <code>SegmentationType</code>	
PH/P/AgeGroupCode	String	Código grupo etario, por ejemplo, ADT	
PH/P/AgeGroup	String	Descripción del grupo etario	
PH/P/DayType	String	Tipo de Día: {HAB, FDS} HAB: día hábil FDS: Fin de semana o feriado	
PH/P/SegmentationType	String	Tipo de segmentación: {NACIONAL, EXTRANJERO} NACIONAL: Precio diferenciado para residente chileno EXTRANJERO: precio diferenciado para Turista	
PH/P/UnitPrice	Decimal	Precio unitario para esta configuración de <code>AgeGroupCode</code> , <code>DayType</code> , <code>SegmentationType</code>	
Dates	Collection (D)	cupos e intervalos por fecha y sus horas	
D/Date	yyyy-MM-dd	Fecha de la cuota	
D/Quotas	Collection (Q)		
D/Q/StartTime	HH:mi:ss	Hora Inicio de la cuota	
D/Q/EndTime	HH:mi:ss	Hora final de la cuota	
D/Q/AvailableQuota	Número	Cantidad de cupos	
D/Q/IsAvailable	Boolean	Indica si existen cupos para realizar una compra	
D/ SalesIntervals	Collection (SI)	Intervalos a mostrar en formato HH:mi (Si <code>UseIntervals</code> es true esta lista se debe presentar	

		para que el usuario seleccione una de ellas, particularmente utilizado para Hop on Hop Off)	
PickupLocations	Collection (PL)	Lugares de recogida	
PL/PickupLocationId	GUID	Identificador del lugar de recogida	
PL/Description	String	Descripción de lugar	
PL/Address	String	Dirección del lugar	
PL/Latitude	Decimal	Coordenada Latitud de lugar	
PL/Longitude	Decimal	Coordenada Longitud del lugar	
PL/PickupTime	HH:mi	Hora y minuto al que el usuario se puede embarcar.	
MeetingPoints	Collection (MP)	Puntos de encuentro donde el pasajero podrá embarcarse	
MP/MeetingPointId	Numero	Identificador del lugar donde podrá embarcar en la excursión	
MP/Description	String	Descripción de lugar	
MP/Address	String	Dirección del lugar	
MP/Latitude	Decimal	Coordenada Latitud de lugar	
MP/Longitude	Decimal	Coordenada Longitud del lugar	
MP/MeetingTime	HH:mi	Hora y minuto al que el usuario se puede embarcar.	

Ejemplo Response

(Notar que para el día 2023-02-02 se configuró un cambio de precio)

```
{
  "tourCode": "982067db-b3af-4e9f-9373-952bc071a051",
  "tourName": "Valparaíso + Viña Del Mar",
  "startTime": "09:30:00",
  "encounterType": "MEETINGPOINT",
  "encounterTypeDescription": "Indica Punto de Encuentro",
  "informationRequired": "REGULAR",
  "informationRequiredDescription": "Información básica de pasajeros y detalle del cliente",
  "useIntervals": false,
  "priceHeaders": [
    {
      "initDate": "2023-01-15",
      "endDate": "2023-02-01",
      "prices": [
        {
          "ageGroupCode": "ADT",
          "ageGroup": "adulto",
          "dayType": "HAB",
          "segmentationType": "NACIONAL",
          "unitPrice": 28000.0
        }
      ]
    },
    {
      "initDate": "2023-02-02",
      "endDate": "2033-02-02",
      "prices": [
        {
          "ageGroupCode": "ADT",
          "ageGroup": "adulto",
          "dayType": "HAB",
          "segmentationType": "NACIONAL",
          "unitPrice": 31000.0
        }
      ]
    }
  ],
  "dates": [
    {
      "date": "2023-02-01",
      "quota": [
```

```
{
  {
    "startTime": "09:30:00",
    "endTime": "15:00:00",
    "availableQuota": 30,
    "isAvailable": true
  }
],
  "salesInterval": [
    "09:30",
    "10:30"
  ]
},
{
  "date": "2023-02-02",
  "quota": [
    {
      "startTime": "09:30:00",
      "endTime": "15:00:00",
      "availableQuota": 30,
      "isAvailable": true
    }
  ],
  "salesInterval": [
    "09:30",
    "10:30"
  ]
}
],
"pickupLocations": [
  {
    "PickupLocationId": "699b6b98-6283-4344-8c19-f6c15730afa6",
    "Description": "Rent A Home",
    "Address": "Santa Magdalena 82, Providencia",
    "Latitude": "-33,4377786081584",
    "Longitude": "-70,6505340905057",
    "TimePickup": "08:30"
  }
],
"meetingPoints": [
  {
    "MeetingPointId": 1,
    "Description": "Plaza de Armas de Santiago",
    "Address": "",
    "Latitude": "-33,4377786081584",
    "Longitude": "-70,6505340905057",
    "TimePickup": "08:30"
  }
]
}
```

3.4 POST /api/v1/rater

Consultar tarifa principalmente utilizado en carrito de compras o al ingresar un cupón.

3.4.1 Request

URI

POST /api/v1/rater

Parámetros

Campo	Tipo Dato	Observación	Información adicional
PaymentMethod	String	Medio de Pago registrados en OzyTrip como por ejemplo Webpay	Opcional
CouponCode	String	Código de Cupón	Opcional
IdBooking	String	Identificador de la reserva. Dado que esta API debe ser utilizado cuando ya existe por lo menos un ítem en carrito este campo es obligatorio.	Obligatorio
Currency	String	Divisa según listado entregado por endpoint /api/v1/tourcodes	Opcional (De no indicar se ocupará CLP)

Ejemplo de llamado (Body):

```
{
  "paymentMethod": null,
  "couponCode": null,
  "idBooking": "RV230825A0401",
  "currency": null
}
```

3.4.2 Response

Campo	Tipo Dato	Observación	Información adicional
TotalAmount	Decimal	Total de la compra	
AdvancePaymentAmount	Decimal	Monto para del anticipo de pago o reserva	
AdvancePaymentPercentage	Decimal	Porcentaje del monto del anticipo expresado como decimal 20% corresponde a 0.2	
DiscountAmount	Decimal	Monto de descuento	
DiscountComment	String	Glosa de descuento global	
Details	Collection (D)	Listado con el detalle	
D/TourCode	String	Código de la excursión	
D/TourName	String	Nombre de la excursión	
D/IdItemEcommerce	String	Identificador interno del eCommerce del ítem del carrito	
D/UnitPrice	Decimal	Precio unitario	

D/CumulativeTotal	Decimal	Total acumulado	
D/DiscountComment	String	Glosa de descuento Particular	
D/DiscountAmount	Decimal	Monto descuento	
D/AgeGroupCode	String	Código del grupo etario	
D/Quantity		Cantidad de pasajero para este grupo etario	
D/DateService	yyyy-MM-ddThh:mi:ss	Fecha de la excursión	

Ejemplo Response

```
{
  "totalAmount": 7990.0,
  "discountAmount": 0.0,
  "AdvancePaymentAmount": 1598.0,
  "AdvancePaymentAmount": 0.2,
  "details": [
    {
      "tourcode": "3cf1ff15-6524-4ebd-bf1e-cfbc97cce729",
      "tourName": "",
      "idItemEcommerce": "ed8103cdf0cb493d8e50f19c33130c4d",
      "unitPrice": 7990.0,
      "cumulativeTotal": 7990.0,
      "discountComment": null,
      "discountAmount": null,
      "ageGroupCode": "ADT",
      "quantity": 1,
      "dateService": "2023-02-16T17:50:00"
    }
  ]
}
```

3.5 POST /api/v1/addToCart

Debe ser utilizado cuando se agrega un ítem al carrito de compras con el fin de realizar la reserva correspondiente de cupos. Entre sus validaciones estará:

- Que queden cupos disponibles para la cantidad solicitada.
- Que este dentro del tiempo para seguir agregando ítems al carrito.

Cada vez que se agrega un ítem al carrito el tiempo se extenderá, indicando un nuevo `BookingExpirationDate`

Se recomienda al llamar a este EndPoint, la colección de `Agegroups` a ocupar, debe ser lo que se tiene actualmente en el carrito del front-end para dicho `IdBooking`, `IdItemEcommerce` y `AgeGroupCode`. Poner especial cuidado en la información especificada para que la información en el carrito sea la misma que se tiene en backend utilizando esta API.

3.5.1 Request

URI

POST /api/v1/addToCart

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip, de no venir se asume que es el primer ítem que se está agregando al carrito del eCommerce, por tanto, devolverá un nuevo <code>IdBooking</code>	Opcional
TourCode	String	Excursión que se desea agregar al carrito	Obligatorio
ServiceDate	yyyy-MM-ddThh:mi:ss	Fecha de la excursión en formato ISO sin zona horaria. Para conformar esta fecha primero el usuario debe seleccionar la fecha, luego con ésta obtener la lista de <code>salesIntervals</code> en <code>/api/v1/tourInformation</code> . Usuario debe seleccionar un elemento de esta lista y conformar la fecha y hora. Solo si <code>UseIntervals</code> es <code>true</code> en <code>/api/v1/tourInformation</code> , sino debe ocupar <code>startTime</code> indicado en API <code>/api/v1/tourInformation</code>	Obligatorio
StartTime	HH:mi	Hora de la excursión seleccionada por el usuario. Listado de <code>salesIntervals</code> en <code>/api/v1/tourInformation</code> , solo si <code>UseIntervals</code> es <code>true</code> en <code>/api/v1/tourInformation</code> . Este valor debe ser el mismo a la hora indicada en <code>ServiceDate</code>	Obligatorio
MeetingPointId	Numero	Identificador del Meeting Point seleccionado	Obligatorio siempre y cuando <code>EncounterType</code> = <code>MEETINGPOINT</code> en <code>/api/v1/tourInformation</code>
PickupLocationId	GUID	Identificado de recogida seleccionado	Obligatorio siempre y cuando <code>EncounterType</code> = <code>PICKUPLOCATION</code> en <code>/api/v1/tourInformation</code>

AgeGroups	Collections (AG)	Listado de IdItemEcommerce, AgeGroupCode y Quantity	Obligatorio
AG/IdItemEcommerce	String	Identificador interno del eCommerce de ítem que se está agregando. Necesario para cuando se necesite remover de la lista	Obligatorio
AG/AgeGroupCode	String	Sigla del grupo etario que se está agregando. Sigla que se puede obtener desde <code>api/v1/tourCodes</code>	Obligatorio
AG/Quantity	Numero	Cantidad del grupo etario debe ser mayor a cero	Obligatorio, mayor a 1

Ejemplo de llamado (Body):

```
{
  "idBooking": null,
  "tourCode": "21e6b389-4878-4485-9168-a48390ad54eb",
  "serviceDate": "2023-08-05T12:00:00",
  "startTime": "12:00:00",
  "meetingPointId": null,
  "pickupLocationId": null,
  "ageGroups": [
    {
      "idItemEcommerce": "e9c02524b83644fdb0e807f50b536688",
      "ageGroupCode": "ADT",
      "quantity": 2
    },
    {
      "idItemEcommerce": "77e0d4b8d98146968dfe35979e677a47",
      "ageGroupCode": "NIN",
      "quantity": 1
    }
  ]
}
```

3.5.2 Response

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id de la Reserva generado en OzyTrip si campo request <code>IdBooking</code> no fue especificado.	
WaitTime	Numero	Tiempo de espera configurado en OzyTrip para este <code>IdBooking</code> donde se indica en minutos	
BookingExpirationDate	yyyy-MM-ddThh:mi:ss	Fecha y hora en que el carrito expira. Toda vez que se agrega un elemento al carrito este campo irá variando.	

Ejemplo Response

```
{
  "idBooking": "RV230731A0101",
  "bookingExpirationDate": "2023-07-31T11:29:47.047",
  "waitTime": 15
}
```

3.5.3 Validaciones

Las siguientes validaciones adicionales se realizarán al momento de ocupar AddToCart:

- Los elementos `MeetingPointId` y `PickupLocationId` siempre deben estar presentes en el json construido.
- Si el `EncounterType` es `NONE` los elementos `MeetingPointId` y `PickupLocationId` deben ir nulos (null)
- Si el `EncounterType` es `MEETINGPOINT` el elemento `MeetingPointId` debe tener un valor entero válido y el `PickupLocationId` debe ir nulo
- Si el `EncounterType` es `PICKUPLOCATION` el elemento `MeetingPointId` debe ir nulo y el `PickupLocationId` deben tener como valor un GUID válido
- Si su correspondiente `IdBooking` ya se encuentra en estado *Registro de pasajeros completado*, es decir, ya se utilizó el endpoint `/api/v1/addPassenger`, se responderá `BadRequest`
- Si `idItemEcommerce` y `ageGroupCode` ya existe en back *OzyTrip*, y se especifica una nueva cantidad (`quantity`), se pisará con el nuevo valor de cantidad indicado en este Request. Se recomienda que al llamar a este EndPoint se haga con la información que se tienen actualmente en el carrito del front-end.
- `quantity` debe ser mayor que cero

Por consiguiente:

- Si el servicio tiene `EncounterType` `MEETINGPOINT` y no viene el elemento `MeetingPointId` o es nulo, entonces el servicio retornará error.
- Si el servicio tiene `EncounterType` `PICKUPLOCATION` y no viene el elemento `PickupLocationId` o es nulo entonces el servicio devuelve un error.
- Si el elemento `MeetingPointId` viene con valor asociado y el servicio tiene un `EncounterType` distinto a `MEETINGPOINT` el servicio retornara un error.
- Si el elemento `PickupLocationId` viene con valor asociado y el servicio tiene un `EncounterType` distinto a `PICKUPLOCATION` el servicio retornara un error.
- Si los elementos `MeetingPointId` y `PickupLocationId` vienen en nulo y el servicio tiene un `EncounterType` distinto a `NONE` el servicio retornara un error.
- Si los elementos `MeetingPointId` y `PickupLocationId` no vienen el servicio retornará error
- No es posible agregar más elementos al carrito si ya se llamó a EndPoint `/api/v1/addPassengers` con los datos de los todos los pasajeros.

3.6 DELETE /api/v1/removeFromCart

Debe ser utilizado cuando se quita un ítem al carrito de compras.

3.6.1 Request

URI

DELETE /api/v1/removeFromCart/{ItemIdEcommerce}/{IdBooking}

Parámetros

Campo	Tipo Dato	Observación	Información adicional
ItemIdEcommerce	String	Identificador interno del eCommerce del ítem que se está quitando.	Obligatorio
IdBooking	String	Id interno de la reserva en OzyTrip.	Obligatorio

Ejemplo de llamado:

```
DELETE http://server:port/api/v1/removeFromCart/5b85c0297/RV230731A0101
```

3.6.2 Response

Al eliminar un ítem del carrito puede suceder que sea el último ítem por tanto IdBooking queda sin efecto donde no podrá ser utilizado nuevamente.

Campo	Tipo Dato	Observación	Información adicional
DeleteType	String	P o T P: Parcial indica que aún quedan ítems en carrito T: Total implica que no quedan ningún ítem en carrito.	

Ejemplo Response

```
{  
  "deleteType": "P"  
}
```

3.6.3 Validaciones

Las siguientes validaciones adicionales se realizarán al momento de ocupar RemoveFromCart:

- Si su correspondiente `IdBooking` ya se encuentra en estado *Registro de pasajeros completado*, es decir, ya se utilizó el endpoint `/api/v1/addPassenger`, se responderá `BadRequest`

3.7 PUT /api/v1/resetTime

Reinicia el tiempo utilizando lo configurado al momento de realizar el llamado a /api/v1/addToCart con el fin de respetar la reserva de los cupos y que estos no sean borrados.

3.7.1 Request

URI

PUT /api/v1/resetTime

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip	Obligatorio

Ejemplo de llamado (Body):

```
{
  "IdBooking": "RV230731A0101",
}
```

3.7.2 Response

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id de la Reserva generado en OzyTrip si campo request <code>IdBooking</code> no fue especificado.	
WaitTime	Numero	Tiempo de espera configurado en OzyTrip para este <code>IdBooking</code> donde se indica en minutos	
BookingExpirationDate	yyyy-MM-ddThh:mi:ss	Fecha y hora en que el carrito expira. Toda vez que se agrega un elemento al carrito este campo irá variando.	

Ejemplo Response

```
{
  "idBooking": "RV230731A0101",
  "bookingExpirationDate": "2023-07-31T11:29:47.047",
  "waitTime": 15
}
```

3.7.3 Validaciones

Las siguientes validaciones adicionales se realizarán al momento de ocupar ResetTime:

- Si su correspondiente `IdBooking` ya se encuentra en estado *Registro de pasajeros completado*, es decir, ya se utilizó el endpoint `/api/v1/addPassenger`, se responderá `BadRequest`

3.8 DELETE /api/v1/booking

En caso de que se desee limpiar carrito para liberar cupos por alguna razón el pago no haya sido exitoso se debe llamar a esta API para limpiar la información. Después de esta acción `IdBooking` queda sin efecto donde no podrá ser utilizado nuevamente.

3.8.1 Request

URI

DELETE /api/v1/booking/{idBooking}

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip.	Obligatorio

Ejemplo de llamado:

```
DELETE http://server:port/api/v1/A00230201
```

3.8.2 Response

HTTP status request: 200 OK o 400 Bad Request dependiendo del éxito o fracaso al realizar el booking. Luego de obtener el OK podrá enviar al cliente a pagar. De fallar no debe permitir enviar al cliente a pagar hasta resolver esta situación.

Ejemplo Response

```
OK
```

3.8.3 Validaciones

Las siguientes validaciones adicionales se realizarán al momento de ocupar Delete Booking:

- Si su correspondiente `IdBooking` ya se encuentra en estado pagado se responderá `BadRequest` indicando que este Booking ya fue pagado.

3.9 POST /api/v1/addPassengers

Esta API complementa los datos del cliente y pasajeros ya ingresados en `/api/v1/addToCart` del carrito de compras, necesarios para realizar el booking final y posterior pago, una vez que se obtenga el OK de esta API.

Este API hará validaciones del ingreso de todos los datos requerido particularmente para excursiones que sea obligatorio ingresar datos de contacto mandatorios.

Una vez ingresado los datos el reloj que controla el tiempo de borrado de las excursiones se detiene y se espera resultado del pago.

3.9.1 Request

URI

POST /api/v1/addPassengers

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip	Obligatorio
Name	String	Nombre del Cliente	Obligatorio
LastName	String	Apellido del Cliente	Obligatorio
Email	String	Email del Cliente	Obligatorio
PhoneNumber	String	Número de teléfono del Cliente	Obligatorio
Country	Sigla según ISO 3166/2	Sigla según norma ISO 3166/2, por ejemplo, para Chile sería CL	Obligatorio
NotificationType	String	Como se entrega los tickets y documentos. {EMAIL, WHATSAPP}	Obligatorio
Passengers	Collection (P)	Colección de pasajeros general que irán las distintas excursiones	Obligatorio
P/IdPassenger	String	Identificador interno eCommerce del pasajero	Obligatorio
P/Name	String	Nombre del Pasajero	Obligatorio Si <code>InformationRequired = REGULAR</code> o <code>MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code>
P/LastName	String	Apellido del Pasajero	Obligatorio Si <code>InformationRequired = REGULAR</code> o <code>MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code>
P/IdentifierType	String	Tipo de Identificador del Pasajero {RUT/DNI/PASS}	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/Identifier	String	Identificador del Pasajero	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/Country	Sigla según ISO 3166/2	Sigla según norma ISO 3166/2, por ejemplo, para Chile sería CL	Opcional
P/PhoneNumber	String	Número de Teléfono del Pasajero	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión

			asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/Email	String	Email del Pasajero (formato correcto del correo debe ser realizado por eCommerce)	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/AgeGroup	String	Grupo etario del pasajero	Obligatorio
ItemsCart	Collection (IC)	ítems del carrito de compra	Opcional (depende de lo que se indique en <code>AnonymousPassengers</code>)
IC/IdItemEcommerce	String	Identificador interno del eCommerce donde ya indicó el <code>tourcode</code> en <code>api/v1/addToCart</code>	Obligatorio
IC/ IdPassenger	String	Identificador interno del pasajero asociado a Item en Carrito (<code>IdItemEcommerce</code>) que finalmente indicará a que excursión asistirá.	Obligatorio

Ejemplo de llamado:

```
{
  "idBooking": "RV230814A0001",
  "name": "Juan",
  "lastName": "Perez",
  "email": "jpperez@gmail.cl",
  "phoneNumber": "+56988889999",
  "country": "br",
  "notificationType": "EMAIL",
  "itemsCart": [
    {
      "idItemEcommerce": "f901b4978dd64d158d0f0fc32edcd82d",
      "idPassenger": "ADT1"
    },
    {
      "idItemEcommerce": "f901b4978dd64d158d0f0fc32edcd82d",
      "idPassenger": "ADT2"
    },
    {
      "idItemEcommerce": "3dfc904867444ce081978ed7731964ed",
      "idPassenger": "NIN1"
    }
  ],
  "passengers": [
    {
      "idPassenger": "ADT1",
      "identifierType": null,
      "identifier": null,
      "name": "Juan",
      "lastName": "Perez",
      "email": null,
      "phoneNumber": null,
      "country": null,
      "ageGroup": "ADT"
    },
    {
      "idPassenger": "ADT2",
      "identifierType": null,
      "identifier": null,
      "name": "Juanita",
      "lastName": "Perez",
      "email": null,
      "phoneNumber": null,
      "country": null,
      "ageGroup": "ADT"
    }
  ]
}
```

```
    },  
    {  
      "idPassenger": "NIN1",  
      "identifierType": null,  
      "identifier": null,  
      "name": "Jose",  
      "lastName": "Perez",  
      "email": null,  
      "phoneNumber": null,  
      "country": null,  
      "ageGroup": "NIN"  
    }  
  ]  
}
```

3.9.2 Response

HTTP status request: 200 OK o 400 Bad Request dependiendo del éxito o fracaso al realizar el booking. Luego de obtener el OK podrá enviar al cliente a pagar. De fallar no debe permitir enviar al cliente a pagar hasta resolver esta situación.

Ejemplo Response

OK

3.10 POST /api/v2/addPassengers

Esta API complementa los datos del cliente y pasajeros ya ingresados en `/api/v1/addToCart` del carrito de compras, necesarios para realizar el booking final y posterior pago, una vez que se obtenga el OK de esta API.

Este API hará validaciones del ingreso de todos los datos requerido particularmente para excursiones que sea obligatorio ingresar datos de contacto mandatorios.

Una vez ingresado los datos el reloj que controla el tiempo de borrado de las excursiones se detiene y se espera resultado del pago.

3.10.1 Request

URI

POST /api/v2/addPassengers

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip	Obligatorio
Name	String	Nombre del Cliente	Obligatorio
LastName	String	Apellido del Cliente	Obligatorio
Email	String	Email del Cliente	Obligatorio
PhoneNumber	String	Número de teléfono del Cliente	Obligatorio
Country	Sigla según ISO 3166/2	Sigla según norma ISO 3166/2, por ejemplo, para Chile sería CL	Obligatorio
NotificationType	String	Como se entrega los tickets y documentos. {EMAIL, WHATSAPP}	Obligatorio
AnonymousPassengers	Boolean	True: Lista de Passengers e ItemsCart Debe venir en vacío ("passengers": []) o en null ("passengers": null) False: Lista de Passengers e ItemsCart Debe venir con sus datos correspondientes.	Obligatorio
Passengers	Collection (P)	Colección de pasajeros general que irán las distintas excursiones	Opcional toda la estructura (depende de lo que se indique en AnonymousPassengers)
P/IdPassenger	String	Identificador interno eCommerce del pasajero	Obligatorio
P/Name	String	Nombre del Pasajero	Obligatorio Si InformationRequired = REGULAR o MANDATORY de la excursión asociada a este IdItemEcommerce
P/LastName	String	Apellido del Pasajero	Obligatorio Si InformationRequired = REGULAR o MANDATORY de la excursión asociada a este IdItemEcommerce
P/IdentifierType	String	Tipo de Identificador del Pasajero {RUT/DNI/PASS}	Obligatorio Si InformationRequired = MANDATORY de la excursión asociada a este IdItemEcommerce o si es marcado como IsResponsible

P/Identifier	String	Identificador del Pasajero	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/Country	Sigla según ISO 3166/2	Sigla según norma ISO 3166/2, por ejemplo, para Chile sería CL	Opcional
P/PhoneNumber	String	Número de Teléfono del Pasajero	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/Email	String	Email del Pasajero (formato correcto del correo debe ser realizado por eCommerce)	Obligatorio Si <code>InformationRequired = MANDATORY</code> de la excursión asociada a este <code>IdItemEcommerce</code> o si es marcado como <code>IsResponsable</code>
P/AgeGroup	String	Grupo etario del pasajero	Obligatorio
ItemsCart	Collection (IC)	ítems del carrito de compra	Opcional (depende de lo que se indique en <code>AnonymousPassengers</code>)
IC/IdItemEcommerce	String	Identificador interno del eCommerce donde ya indicó el <code>tourcode</code> en <code>api/v1/addToCart</code>	Obligatorio
IC/ IdPassenger	String	Identificador interno del pasajero asociado a Item en Carrito (<code>IdItemEcommerce</code>) que finalmente indicará a que excursión asistirá.	Obligatorio

Ejemplo de llamado:

```
{
  "anonymousPassengers": false
  "idBooking": "RV230814A0001",
  "name": "Juan",
  "lastName": "Perez",
  "email": "jpperez@gmail.cl",
  "phoneNumber": "+56988889999",
  "country": "br",
  "notificationType": "EMAIL",
  "itemsCart": [
    {
      "idItemEcommerce": "f901b4978dd64d158d0f0fc32edcd82d",
      "idPassenger": "ADT1"
    },
    {
      "idItemEcommerce": "f901b4978dd64d158d0f0fc32edcd82d",
      "idPassenger": "ADT2"
    },
    {
      "idItemEcommerce": "3dfc904867444ce081978ed7731964ed",
      "idPassenger": "NIN1"
    }
  ],
  "passengers": [
    {
      "idPassenger": "ADT1",
      "identifierType": null,
      "identifier": null,
      "name": "Juan",
      "lastName": "Perez",
      "email": null,
      "phoneNumber": null,
      "country": null,
      "ageGroup": "ADT"
    }
  ]
}
```

```
    },
    {
      "idPassenger": "ADT2",
      "identifierType": null,
      "identifier": null,
      "name": "Juanita",
      "lastName": "Perez",
      "email": null,
      "phoneNumber": null,
      "country": null,
      "ageGroup": "ADT"
    },
    {
      "idPassenger": "NIN1",
      "identifierType": null,
      "identifier": null,
      "name": "Jose",
      "lastName": "Perez",
      "email": null,
      "phoneNumber": null,
      "country": null,
      "ageGroup": "NIN"
    }
  ]
}
```

3.10.2 Response

HTTP status request: 200 OK o 400 Bad Request dependiendo del éxito o fracaso al realizar el booking. Luego de obtener el OK podrá enviar al cliente a pagar. De fallar no debe permitir enviar al cliente a pagar hasta resolver esta situación.

Ejemplo Response

```
OK
```

3.11 POST /api/v1/pay

Una vez que el cliente haya pagado exitosamente se debe llamar a esta API para registrar el pago y tomar los cupos de forma definitiva.

3.11.1 Request

URI

POST /api/v1/pay

Parámetros

Campo	Tipo Dato	Observación	Información adicional
IdBooking	String	Id interno de la reserva en OzyTrip	Obligatorio
TotalAmount	Decimal	Monto total de la compra. Debe ser el mismo con el que se ocupó en /api/v1/startPayment/ campo Total	Obligatorio
HasAdvancePayment	Boolean	Indica si el cliente esta pagando sólo el Pago Anticipado. Debe ser el mismo con el que se ocupó en /api/v1/startPayment/ campo AdvancePaymentAmount	Obligatorio
PaymentDate	yyyy-MM-ddThh:mi:ss	Fecha en que se realiza el pago.	Obligatorio
Currency	String	Divisa según listado entregado por endpoint /api/v1/tourcodes	Opcional (De no indicar se ocupará CLP)
CardType	String	Tipo tarjeta. {VISA, MASTERCARD, AMERICAN EXPRESS, DINNERS CLUB, REDCOMPRA}	Opcional
CardNumber	Numero(4)	últimos cuatro dígitos de la tarjeta	Opcional
AuthorizationTransactionId	String	Identificador de la transacción en el medio de pago	Obligatorio
PaymentMethod	String	Medio de Pago como Webpay, Paypal, Ebanx, Stripe, etc	Obligatorio
IdOrderNumber	String	Identificador del Número de Orden. En Woocommerce corresponde al Nro de Pedido	Obligatorio
Cupon	String	Si se utiliza algún cupón en esta compra, éste debe ser indicado en este campo	Opcional

Ejemplo de llamado:

```
{
  "idBooking": "RV230811A1001",
  "totalAmount": 98000.0,
  "couponCode": null,
  "hasAdvancePayment": false,
  "paymentDate": "2023-08-11T13:40:15",
  "currency": null,
  "cardType": null,
  "cardNumber": null,
  "authorizationTransactionId": "1715798498",
  "paymentMethod": "W",
  "idOrderNumber": "c638b9b526be4db49c3f8a3f89bcf8d9"
}
```

3.11.2 Response

Campo	Tipo Dato	Observación	Información adicional
SalesCode	String	Código de venta	Opcional cuando HasAdvancePayment = true
IdBooking	String	Id de la Reserva enviado a OzyTrip.	Obligatorio
HasAdvancePayment	Boolean	Indica si esta venta se pagó como anticipo (true) o solo como anticipo (false)	Obligatorio
Balance	Numérico	Saldo por pagar en caso de HasAdvancePayment este en true.	Obligatorio

Ejemplo Response

```
{
  "salesCode": "VT230811A022701",
  "idBooking": "RV230811A1001",
  "HasAdvancePayment": false,
  "balance": 0
}
```

3.11.3 Validaciones

- Si el `idBooking` enviado se encuentra con sus pasajeros registrados, el sistema le responderá 200 OK con la estructura JSON indicada en 3.11.2 Response
- Si el `idBooking` enviando ya se encuentra pagada el sistema responderá 200 OK con la estructura JSON indicada en 3.11.2 Response. Vale decir que se podrá utilizar este método para recuperar el `salesCode` correspondiente en caso de necesitar dicha información.
- En cualquier otro caso responderá 400 Bad Request con el detalle del error.
- En caso de existir una doble llamada, esto es, llamar dos veces casi al mismo tiempo, tenemos dos escenarios:
 - Si la venta no ha sido registrada, uno de los hilos de ejecución será descartado respondiendo 409 Conflict Error y el otro la respuesta 200 OK con la estructura JSON indicada en 3.11.2 Response.
 - Si la venta ya ha sido registrada con anterioridad ambos hilos de ejecución responderán 200 OK con la estructura JSON indicada en 3.11.2 Response.