

Al momento de inicializar el formulario se hace la búsqueda de las tarifas según el hotel seleccionado

se deben de seleccionar los datos de check in, check out, adultos, menores y las edades.

The screenshot shows a web form titled "Hotel Availability Calendar". It contains two dropdown menus at the top: "Select a hotel" with "Nacional Inn" selected, and "Select a rate plan" with "Commissioned Mexican peso fare" selected. Below these are five input fields: "Check In" with a calendar icon and date "2025-10-01", "Check Out" with a calendar icon and date "2025-10-05", "Adults" with a numeric input "1", "Childs" with a numeric input "0", and "Childs Ages" with an empty text box. At the bottom are two buttons: "Search Availability" in blue and "second search" in orange.

Una vez con estos datos implementados se procede a hacer la búsqueda de los cuartos, los cuales son mostrados en una lista junto a su información, en este caso el nombre la descripción y su precio

This screenshot shows the same form as above, but with an additional section titled "Available Rooms" below the search buttons. This section lists three room options, each with a description, a price, and a "Book Now" button.

Room Name	Description	Price	Action
1036385 - Suite	Apartment has 1 double bed. Accommodation in the annex building, which allows you to enjoy all of our themed common areas, and which brings modernity and a sense of freedom to its interior with its all-glass façade.	\$: 480 USD	Book Now
1036388 - Premium Triple	Apartment has 01 double bed + 01 single bed (accommodation inside the castle). Accommodations feature a 32" LED cable TV, minibar, a private bathroom with a soaking tub and free toiletries. Wi-Fi is free in all areas of the hotel. Some accommodations include a balcony. All accommodations have views of the natural beauty of Campos do Jordão.	\$: 1080 USD	Book Now
1036387 - Double Luxury	Accommodation in the annex building, which provides the opportunity to use our themed common areas, and which brings modernity and a sense of freedom to its interior with its all-glass façade.	\$: 880 USD	Book Now

Pax contact details

Contact for Pax #1

aaaaa

bbbb

cccc@asaed.com

8127638712

Send Hotel Reservation

Commit Reservation

Reservation ID

RES000486-198336

Transaction ID

29521f80-651c-1c96-9542-b35ef61889be

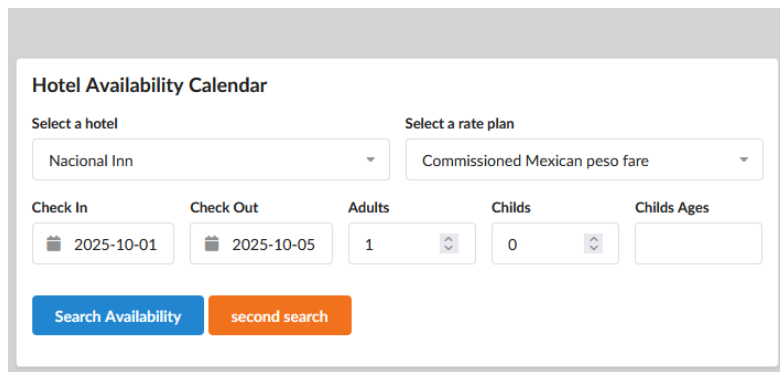
Commit Reservation

Estado	Método	URL	Iniciador	Tipo	Transferido	T.	Encabezados	Cookies	Solicitud	Respuesta	Tiempos	Seguimiento de pila	Se...
200	GET	https://omnibeas.dubslaris.com/Omnibeas/HotelSearch?api_key=YcWhP...	full_text-81 (fetch)	json	15.26 kB	6.9	Propiedades del filtro						
200	GET	data:application/font-woff;base64,d09GRk9uVEEAAASEAAIAAAABwQAAQAL...	font	font-woff	0 B	15.5	JSON						
200	GET	https://cdnjs.cloudflare.com/ajax/libs/omantic-ui/2.9.4/themes/default/ass...	font	octet-stream	79.28 kB	78.	Moreindicator: false						
200	GET	https://omnibeas.dubslaris.com/Omnibeas/GetHotelRatePlans?api_key=y...	full_text-882 (fetch)	json	8.60 kB	202	TotalRecords: null						
200	POST	https://omnibeas.dubslaris.com/Omnibeas/GetHotelAvail	full_text-518 (fetch)	json	5.98 kB	33.	PageNumber: null						
200	POST	https://omnibeas.dubslaris.com/Omnibeas/SendHotelResEstimate	full_text-737 (fetch)	json	4.43 kB	11.	MaxResponses: null						
200	POST	https://omnibeas.dubslaris.com/Omnibeas/SendHotelResCommit	full_text-835 (fetch)	json	1.26 kB	488	EchoToken: "system_Z954316-d021-e7cd-4fcd-24e704802eb"						
200	GET	https://omnibeas.dubslaris.com/Omnibeas/GetHotelRatePlans?api_key=y...	full_text-483 (fetch)	json	5.37 kB	42.	TimeStamp: "2025-09-03T14:01:43.5294879Z"						
200	POST	https://omnibeas.dubslaris.com/Omnibeas/GetHotelAvail	full_text-518 (fetch)	json	1.60 kB	1.3	Target: 0						
200	GET	https://omnibeas.dubslaris.com/Omnibeas/GetHotelRatePlans?api_key=y...	full_text-483 (fetch)	json	7.14 kB	12.	Version: 1.2000000000000000017635639402504646778106689453125						
200	POST	https://omnibeas.dubslaris.com/Omnibeas/GetHotelAvail	full_text-518 (fetch)	json	5.93 kB	40.	PrimaryLangID: 1						
							Success: []						
							WarningsType: null						
							ErrorType: null						
							POS: null						
							HotelReservationsType: null						
							ApiResponseType: "Committed"						
							RetransmissionIndicator: null						
							TransactionIdentifier: "29521890-631c-1c96-9542-635ef61889be"						
11 solicitudes 657.62 kB / 134.85 kB transferido Finalizar: 6:06 min													
Financiar													

Pasos para generar una reserva mediante la integración del api de Omnibees

El primer paso es buscar un hotel

```
$settings = $settings['settings']['logger_omnibees'];  
$logger = new Monolog\Logger($settings['name']);  
$logger->pushProcessor(new Monolog\Processor\UidProcessor());  
$logger->pushHandler(new Monolog\Handler\StreamHandler($settings['path'], $settings['level']));  
  
$omnibees = new \APIS\Omnibees($logger);  
  
$response = $omnibees->HotelSearch();
```



The screenshot shows a web form titled "Hotel Availability Calendar". It contains several input fields and buttons. At the top, there are two dropdown menus: "Select a hotel" with "Nacional Inn" selected, and "Select a rate plan" with "Commissioned Mexican peso fare" selected. Below these are five input fields: "Check In" with a calendar icon and the date "2025-10-01", "Check Out" with a calendar icon and the date "2025-10-05", "Adults" with a numeric input "1" and a dropdown arrow, "Childs" with a numeric input "0" and a dropdown arrow, and "Childs Ages" which is empty. At the bottom, there are two buttons: a blue "Search Availability" button and an orange "second search" button.

Es identificado el hotel que se quiere utilizar se debe inicializar la búsqueda de fechas disponibles, con los siguientes pasos:

1. Crear el objeto cuarto, parametrizando las fechas
2. Crear los huéspedes
3. Agregar los huéspedes al cuarto
4. Configurar los datos para la búsqueda del cuarto, en caso de no insertarlos se manejarán los datos por defecto o se arrojará un error al no encontrar algún dato obligatorio
 - Rate per Market
 - Código del hotel
 - Id de tarifa

Hotel Availability Calendar

Select a hotel

Nacional Inn

Select a rate plan

Commissioned Mexican peso fare

Check In

2025-10-01

Check Out

2025-10-05

Adults

1

Childs

0

Childs Ages

Search Availability

second search

Available Rooms

1036385 - Suite

Apartment has 1 double bed. Accommodation in the annex building, which allows you to enjoy all of our themed common areas, and which brings modernity and a sense of freedom to its interior with its all-glass façade.

\$ 480 USD

Book Now

1036388 - Premium Triple

Apartment has 01 double bed + 01 single bed (accommodation inside the castle). Accommodations feature a 32" LED cable TV, minibar, a private bathroom with a soaking tub and free toiletries. Wi-Fi is free in all areas of the hotel. Some accommodations include a balcony. All accommodations have views of the natural beauty of Campos do Jordão.

\$ 1080 USD

Book Now

1036387 - Double Luxury

Accommodation in the annex building, which provides the opportunity to use our themed common areas, and which brings modernity and a sense of freedom to its interior with its all-glass façade.

\$ 880 USD

Book Now

deberá

Hotel Availability Calendar

Select a hotel

City Garden Hotels

Select a rate plan

Tarifa Mexican peso comisionada

Check In

2025-10-01

Check Out

2025-10-05

Adults

1

Childs

0

Childs Ages

Search Availability

second search

Available Rooms

1036454 - Superior

Apartamentos confortáveis com área de 27m², duas camas de solteiro, ficam entre 1º e o 9º andar.

\$ 756 USD

Book Now

1036456 - Suite

Apartamentos espaçosos, com 40m², ficam entre o 1º ao 9º andar, possuem dois ambientes com sala e copa. Os quartos tem cama de casal e sofá cama, e ainda contam com estação de trabalho, conexão para internet, ferro elétrico, tábua de passar roupa e isolamento acústico.

\$ 945 USD

Book Now

1036457 - Standard

Apartamentos espaçosos, com 40m², ficam entre o 1º ao 9º andar, possuem dois ambientes com sala e copa. Os quartos tem cama de casal e sofá cama, e ainda contam com estação de trabalho, conexão para internet, ferro elétrico, tábua de passar roupa e isolamento acústico.

\$ 754 USD

Book Now

1036455 - Luxe

Apartamentos acolhedores com área de 27m², cama de casal, ficam entre 10º ao 22º andar. Os quartos possuem estação de trabalho, conexão para internet, ferro elétrico, tábua de passar roupa e isolamento acústico.

\$ 567 USD

Book Now

5. Se

obtener

de

```
"last_name" => "Doe",
"email" => "john.doe@example.com",
"phone" => "123-456-7890",
"PrimaryIndicator" => false,
"AgeQualifyingCode" => "Adult",
"ResGuestRPH" => 2,
]);

$room->addGuest($guest2);

$room->setISOMarket("CHE");
$room->setResortId(198336);
$room->setRatePlan(3236772);

$data = $room->availableArray();

$response = $omnibees->GetHotelAvail($data,[$room]);
```

el array con la estructura de búsqueda y mandarlo la api Omnibees

Los parámetros de la función GetHotelAvail Permiten realizar la búsqueda de una reserva nueva tanto como realizar la búsqueda para una modificación de cuarto.

```
/**
 * Obtiene la disponibilidad de habitaciones para un hotel en un rango de fechas
 * @param string $cin Fecha de entrada
 * @param string $cout Fecha de salida
 * @param int $hotel_code Código del hotel
 * @param string $rate_plan Código del plan de tarifas
 * @param string $room_id Código de la habitación
 * @param string $ISOMarket Código del mercado ISO
 * @param int $adults Número de adultos
 * @param int $childs Número de niños
 * @return string Respuesta de la API
 */
function GetHotelAvail($data,$rooms,$isModify = false,$reservationNumber = null,$indexNumber = null)
```

1. Se carga en memoria el json de búsqueda inicial de cuartos
2. se agregan datos generales como el UID , el time stamp y la versión de la api a utilizar .
3. se agregan los datos de la reserva como el código de hotel, fechas de check in y check out, tarifa, el id del rate market,
 - en caso de existir el filtro de tarifa este será aplicado
4. se procesan cada uno de los cuartos mandados en el segundo parámetro para agregarlos en la petición de búsqueda .
5. en caso de ser una modificación se agrega el número de reservación y los datos pertinentes a la búsqueda.

```
function GetHotelAvail($data,$rooms,$isModify = false,$reservationNumber = null,$indexNumber = null){
    $string = file_get_contents(self::REQUEST_FOLDER . "HotelAvailRQ.json");

    $json_a = json_decode($string);
    $uuid = $this->generateGUID();
    // lowercase the uuid
    $uuid = strtolower($uuid);
    $json_a->EchoToken = "system_" . $uuid;
    $json_a->TimeStamp = date("c");
    $json_a->Version = "7.2";
    $json_a->HotelSearchCriteria->Criterion->HotelRefs[0]->HotelCode = intval($data["hotel_code"]);
    $json_a->HotelSearchCriteria->Criterion->StayDateRange->Start = $data["cin"] . "T00:00:00";
    $json_a->HotelSearchCriteria->Criterion->StayDateRange->End = $data["cout"] . "T00:00:00";

    if(!empty($data["rate_plan"])):
        if(!is_array($data["rate_plan"]))
            $data["rate_plan"] = [$data["rate_plan"]];
        $json_a->HotelSearchCriteria->Criterion->TPA_Extensions->RatePlanIds = $data["rate_plan"];
    endif;

    if(!empty($data["room_id"])):
        // $json_a->HotelSearchCriteria->Criterion->RoomStayCandidatesType->RoomStayCandidates[0]->RoomID = intval($room_id);
        $json_a->HotelSearchCriteria->Criterion->TPA_Extensions->RoomTypeIds = [intval($data["room_id"]);];
    endif;

    if(!empty($data["ISOMarket"]))
        $json_a->POS->Sources[0]->ISOCountry = $data["ISOMarket"];

    if($data["ISOMarket"] == "ALL" || $data["ISOMarket"] == NULL)
        $json_a->POS = NULL;

    foreach($rooms as $index => $room):
        $room_schema = $room->getSchema();
        $room_schema->RPH = $index;
        $json_a->HotelSearchCriteria->Criterion->RoomStayCandidatesType->RoomStayCandidates[] = $room_schema;
    endforeach;

    if($isModify):
        $ProfileInfos =
        [
            "UniqueID" => [
                "ID" => $reservationNumber,
                "Type" => 14
            ]
        ];
        $json_a->HotelSearchCriteria->Criterion->ProfilesType = new stdClass();
        $json_a->HotelSearchCriteria->Criterion->ProfilesType->ProfileInfos = [$ProfileInfos];
        $json_a->HotelSearchCriteria->Criterion->RoomStayCandidatesType->RoomStayCandidates[0]->BookingCode = $indexNumber;
        $json_a->IsModify = true;
    endif;

    $body = json_encode($json_a);
```

- 6 Una vez concluido el proceso de configuración de los datos para la búsqueda se procede a hacer la codificación de los mismos al formato json .
- 7 son enviados los datos asegurándose de almacenar un que contiene la información enviada como la información recibida y detalles relevantes para hacer depuración y corrección de errores

```
$body = json_encode($json_a);

$request = new Request('POST', $this->apiUrl . '/api/Pull/GetHotelAvail', $this->headers, $body);
$res = $this->client->sendAsync($request)->wait();
$content = $res->getBody()->getContents();
// Log the request and response
$this->logger->info("Petition",
[
    "incoming"=> [
        "function" => "GetHotelAvail",
        "Method" => "direct",
        "api" => "",
        "Route" => "/api/Pull/GetHotelAvail",
        "Data" => json_decode($body,true),
    ],
    "response" =>[
        "status" => $res->getStatusCode(),
        "type" => $res->getHeaderLine('Content-Type'),
        "body" => json_decode($content,true)
    ]
]);

return $content;
```

- Una vez que el proceso de búsqueda de cuartos ha sido exitoso se procede a llamar el método que devuelven los datos para iniciar el proceso de reserva.
- se mandan los cuartos junto al dato del market source y los huéspedes
- Datos extra como el tipo de pago se mandan por separado, en un objeto que contiene estas configuraciones especiales de la reserva

```
$room->setISOMarket("CHE");
$room->setResortId(198336);
$room->setRatePlan(3236772);
$room->setRoomId(1036387);
$ISOMarket = "AUS";
$rooms = [
    $room->reservationStartArray()
];

/*
a) DirectBill = 2
b) CreditCard = 5
c) Voucher = 3
d) PrePay = 4
*/

$response = $omnibees->SendHotelResInitiate($rooms, $ISOMarket,[$guest1, $guest2],false,[
    "PaymentType" => 3
]);
```

La función SendHotelResInitiate usa la información de los cuartos huéspedes código de mercado para rellenar cada uno de los datos necesarios dentro de la plantilla json; se realiza se realiza en validaciones de datos obligatorios para continuar con el proceso de reserva. Una vez pasadas estas validaciones se inyecta la información de cada uno de los cuartos a cotizar dentro del objeto.

```
function SendHotelResInitiate($rooms, $ISOMarket = "MEX", $guests, $isModify = false, $reservation_data = []) {  
    if(empty($rooms) || !is_array($rooms))  
        throw new \InvalidArgumentException("Invalid rooms data");  
    if(empty($guests) || !is_array($guests))  
        throw new \InvalidArgumentException("Invalid guests data");  
    $string = file_get_contents(self::REQUEST_FOLDER . "SendHotelRes_Initiate_RQ.json");  
    $json_a = json_decode($string);  
    $uuid = $this->generatedUUID();  
    // lowercase the uuid  
    $uuid = strtolower($uuid);  
    $json_a->EchoToken = "Mytest." . $uuid;  
    $json_a->TimeStamp = date("c");  
    foreach($guests as $index => $guest):  
        $guest->setBPN($index + 1);  
        $json_a->HotelReservationsType->HotelReservations[0]->ResRequestsType->ResRequests[] = $guest->getSchema();  
    endforeach;  
    foreach($rooms as $index => $room):  
        if(empty($room["RoomID"])) throw new \Exception("RoomID is required for room {$index}")  
        if(empty($room["GuestCounts"])) throw new \Exception("GuestCounts are required for room {$index}");  
        if(empty($room["StartDate"])) throw new \Exception("StartDate is required for room {$index}");  
        if(empty($room["EndDate"])) throw new \Exception("EndDate is required for room {$index}");  
        if(empty($room["HotelCode"])) throw new \Exception("HotelCode is required for room {$index}");  
        if(empty($room["RatePlanID"])) throw new \Exception("RatePlanID is required for room {$index}");  
        $cin = $room["StartDate"];  
        $cout = $room["EndDate"];  
        $hotel_code = $room["HotelCode"];  
        $rate_plan_id = $room["RatePlanID"];  
        $room_id = $room["RoomID"];  
        // add a room stay  
        $basic_room_stay = json_decode(file_get_contents(self::SCHEMAS_FOLDER . "RoomStay.json"));  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[] = $basic_room_stay;  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[0]->CommentsType->Comments = [];  
        // add a comment  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[0]->CommentsType->Comments[] = (object)[  
            "Description" => "This is my {$index} room comment",  
            "Language" => null,  
            "Name" => null  
        ];  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->BasicPropertyInfo->HotelRef->HotelCode = intval($hotel_code);  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->RoomRates[0]->RatePlanID = intval($rate_plan_id);  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->RoomRates[0]->RoomID = intval($room_id);  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->RoomRates[0]->EffectiveDate = $cin . "T00:00:00";  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->RoomRates[0]->ExpireDate = $cout . "T00:00:00";  
        $json_a->HotelReservationsType->HotelReservations[0]->RoomStaysType->RoomStays[$index]->GuestCountsType->GuestCounts = $room["GuestCounts"];  
    endforeach;  
    if($isModify):
```

Checkout

resortCode

198336

roomCode

1036385

RatePlanID

3236715

checkIn

2025-10-01

checkOut

2025-10-05

adults

1

children

0

children_ages

children_ages

Pax contact details

Contact for Pax #1

aaaaa

bbbb

cccc@asaed.com

8127638712

Send Hotel Reservation

En caso de ser una modificación se agregan los datos necesarios para la búsqueda

```
if($isModify):
    $ProfileInfos =
    [
        "UniqueID" => [
            "ID" => $reservation_data["reservationNumber"],
            "Type" => 14
        ]
    ];

    $json_a->HotelReservationsType->HotelReservations[0]->ProfilesType = new stdClass();
    $json_a->HotelReservationsType->HotelReservations[0]->ProfilesType->ProfileInfos = $ProfileInfos;
    $json_a->HotelReservationsType->HotelReservations[0]->RoomStayCandidatesType->RoomStayCandidates[0]->BookingCode = $reservation_data["IndexNumber"];
endif;
```

Se agregan los datos faltantes para completar la petición y se validan algunas configuraciones extra de la reserva como el método de pago; posterior a esto se procede a hacer la petición y guardará los correspondiente

```
$json_a->HotelReservationsType->HotelReservations[0]->ResGlobalInfo->Guarantee->GuaranteesAcceptedType->GuaranteesAccepted[0]->GuaranteeTypeCode = 2; // credit card as default

if(!empty($reservation_data["PaymentType"]))
    $json_a->HotelReservationsType->HotelReservations[0]->ResGlobalInfo->Guarantee->GuaranteesAcceptedType->GuaranteesAccepted[0]->GuaranteeTypeCode = $reservation_data["PaymentType"];

$json_a->TransactionIdentifier = $uuid;
$json_a->Version = "7.2";
$body = json_encode($json_a);

$request = new Request('POST', $this->apiURL . '/api/Pull/SendHotelRes', $this->headers, $body);
$res = $this->client->sendAsync($request)->wait();
$content = $res->getBody()->getContents();
// Log the request and response
$this->logger->info("Petition",
[
    "incoming"=> [
        "function" => "SendHotelRes",
        "method" => "direct",
        "api" => "",
        "route" => "/api/Pull/SendHotelRes",
        "data" => json_decode($body,true),
    ],
    "response" => [
        "status" => $res->getStatusCode(),
        "type" => $res->getHeaderLine('Content-Type'),
        "body" => json_decode($content,true)
    ]
]);
return $content;
}
```

Una vez que se ha mandado la petición a la reserva y se confirma que se ha generado de manera exitosa se procede a hacer el commit, Usando los datos del código de reserva y el id de la transacción

```
$reservationId = "RES000481-198336";
$transactionId = "ab258d87-4923-3b20-938a-22dc87a7b8c8";

$response = $omnibeas->SendHotelResCommit($reservationId, $transactionId);
// ----- termina la reserva original -----
```

Commit Reservation

Reservation ID

RES000486-198336

Transaction ID

29521f80-651c-1c96-9542-b35ef61889be

Commit Reservation

Dentro de la función que manda el cómic se anexan los datos del código de reserva y la transacción.

```
function SendHotelResCommit($reservation_id, $transaction_id){
    $string = file_get_contents(self::REQUEST_FOLDER . 'SendHotelRes_Commit_RQ.json');
    $json_a = json_decode($string);
    $uuid = $this->generateGUID();
    $uuid = strtolower($uuid);

    // Set the EchoToken, TimeStamp, TransactionIdentifier and UniqueID
    $json_a->EchoToken = "system_" . $uuid;
    $json_a->TimeStamp = date("c");
    $json_a->TransactionIdentifier = $transaction_id;
    $json_a->HotelReservationsType->HotelReservations[0]->UniqueID->ID = $reservation_id;

    $body = json_encode($json_a);
    $request = new Request('POST', $this->apiUrl . '/api/Pull/SendHotelRes', $this->headers, $body);
    $res = $this->client->sendAsync($request)->wait();
    $content = $res->getBody()->getContents();
    // Log the request and response
    $this->logger->info("Petition",
    [
        "incoming"=> [
            "function" => "SendHotelResCommit",
            "Method" => "direct",
            "api" => "",
            "Route" => "/api/Pull/SendHotelResCommit",
            "Data" => json_decode($body,true),
        ],
        "response" =>[
            "status" => $res->getStatusCode(),
            "type" => $res->getHeaderLine('Content-Type'),
            "body" => json_decode($content,true)
        ]
    ]
    );
    return $content;
}
```