

# MyGuest Public Api

## Introduction

Welcome to the BookingCenter MyGuest API Partner Documentation! The APIs documented here allow you to programmatically:

- Create a Request
- Manage a Request
- Receive Request Information
- Receive Staff Information
- Receive Form Information
- Receive Category Information

## Terminology

The following terms are used throughout API documentation:

- **BookingCenter Partner Manager** — Your Partner Manager helps get you set up to use the API.
- **Property** — The property the **Partner** will be accessing or managing information for. The **property** grants MyGuest API access to the **Partner** and their **Partner Application**.
- **Partner Application** — The software developed by **Partner** that consumes the MyGuest APIs to manage listings on a host's behalf.
- **Partner** — Develops and manages the Partner application. Partners are the target audience for this documentation.

## Basic API Usage Steps

- Your BookingCenter Partner Manager approves your application for use and will provide you with your Api Credentials.
- The Property grants the Partner Application permission to manage their property account via the API.
- Your application should now be set up and, depending on authorization and permissions, be able to:
  - Create and update requests (requests)
  - Retrieve a list of available properties (properties) that have authorized partner application access.
  - Access available request forms (forms) for the specified property.
  - Access available staff (staff) for the specified property.
  - Access available categories (categories) for the specified property.

## API Development Cycle

- Acceptance into the Api Partner Program
- Partner Manager provides credentials to development environment.
- Partner completes development and testing against development environment.
- Partner Manager provides credentials for production environment.
- Property grants the Partner Application permission to access their property account via the API.

## Authentication:

All endpoints require authentication in the form of a login and key pair that were provided upon Partner registration. The `api_login` can be defined as either a string, or an integer, however the `api_key` must always be in UUID format.

it must be included in the request headers

```
x-api-login: 1123457
x-api-key: f4f7ca6c-72b7-4b1c-921e-f31243c2b5ce
```

Failure to include valid credentials as described will return a http status code 401 with a response *Unauthorized*.

## Rate Limiting

To ensure API stability and prevent abuse, BookingCenter enforces rate limits at the partner level. Your application is responsible for operating within these rate limits. Once a rate limit has been exceeded, no more requests are handled until the limit expires. Attempting to issue requests repeatedly beyond the limit could result in your application being blocked for a longer period, other legitimate requests being rejected, and/or the IP address being blacklisted.

If you encounter issues with rate limits for your application, please ensure you have taken the following or similar actions:

- Build a queuing system to ensure that you don't exceed the limits.

- Ensure that you do exponential backoff if you do exceed a rate limit.
- Ensure that the you have the `x-api-login` and `x-api-key` headers on your requests.

In the event the application exceeds the API Rate limiting an 429 error will be returned with a ``retry-after`` element notifying the application after the UTC time in which the API will accept new requests from it. A sample 429 rate limit message is provided below.

```
{
  "api_version": 3,
  "status": "failed",
  "errors": {
    "code": 429,
    "message": "Too many requests",
    "retry-after": "2020-02-11T02:44:47+00:00"
  },
  "request_uuid": "b84ecfde-19de-4ede-a96a-91f86c623841"
}
```

## Validation:

## Technical Support

To assist partner integration to our API platform, please use our support system at our [Online Support Portal](#). Every API call response contains meta information including the status of the response, `api_version` and the `request_uuid`. When requesting technical support please provide the endpoint you are making the call, your formatted request and the full response from the api including the `request_uuid`.

Sample Success meta information:

```
"api_version": 3,
"status": "success",
"request_uuid": "35f52462-a3e6-4a37-aaed-c7326932fb3a"
```

Sample Error meta information:

```
{
  "api_version": 3,
  "status": "failed",
  "errors": {
    "status": 404,
    "title": "record not found",
    "details": "no record 345 found"
  },
  "request_uuid": "b0370ef0-3c83-40af-a993-0201f37c33dc"
}
```

## Endpoints

### Properties

#### Retrieve All Properties

This call returns all of a properties that have authorized the provider access via the API.

Fetch all Properties as Properties Object array.

```
GET v3/property
```

#### Request URL Parameters

Parameter	Description
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null

#### Example Response

```
{
  "properties": [
    {
      "id": 2,
      "name": "A MyPMS Demo Hotel",
      "site_url": "http://Hotel.bookingcenter.com",
      "timezone": "America/Dawson",
      "contact_email": "mypms@demohotel.com"
    },
    {
      "id": 3,
      "name": "Steves Test Property",
      "site_url": "http://Hotel.bookingcenter.com",
      "timezone": "Africa/Abidjan",
      "contact_email": null
    },
    {
      "id": 10,
      "name": "Lava Ridges Test Property",
      "site_url": "www.lavaridge.org",
      "timezone": "America/Dawson",
      "contact_email": "lavaridges@email.com"
    }
  ],
  "api_version": 3,
  "status": "success",
  "request_uuid": "74ec482c-5a37-416b-820e-0c0e52f0a6e7"
}
```

## Retrieve Single Property

This call returns all of a properties that have authorized the provider access via the API. Properties are returned in a paginated collection which provides a Meta Array with pagination information.

Fetch all Properties as Properties Object array.

```
GET v3/property/:id
```

#### Request URL Parameters

Parameter	Description
<b>id</b> integer required	The id of the property being retrieved

## Forms

## Retrieve All Forms

This call returns all of a properties forms and the fields associated with each form. Forms are returned in a paginated collection which provides a Meta Array with pagination information.

Fetch a Properties forms as Form Object array.

```
GET  v3/forms
```

### Request URL Parameters

Parameter	Description
<div>hotel_id</div> <div>integer</div> <div>required</div>	Property Numerical identifier Cannot be null
<div>limit</div> <div>integer</div> <div>optional, default is 25</div>	The maximum number of forms to return, up to 50.

## Retrieve Single Form

This call returns a specific form and the form fields associated for the requested.

Fetch a specific form as Form Object array.

```
GET  v3/forms/:form_id
```

### Request URL Parameters

Parameter	Description
<div>id</div> <div>integer</div> <div>required</div>	The id of the form
<div>hotel_id</div> <div>integer</div> <div>required</div>	Property Numerical identifier Cannot be null

## Staff

### Retrieve All Staff

This call returns all available staff by property and their roles. Forms are returned in a paginated collection which provides a Meta Array with pagination information.

Fetch a Properties staff as Staff Object array.

```
GET  v3/staff
```

### Request URL Parameters

Parameter	Description
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null

## Retrieve Specific Staff

This call returns all available staff by property and their roles.

Fetch a Properties staff as Staff Object array.

```
GET v3/staff/:username
```

### Request URL Parameters

Parameter	Description
<b>username</b> string required	Specific staff username obtained from the retrieve all staff call
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null

## Categories

### Retrieve All Categories

This call returns all available form categories including their respective department. Categories are returned in a paginated collection which provides a Meta Array with pagination information.

Fetch a Properties categories as categories Object array.

```
GET v3/categories
```

### Request URL Parameters

Parameter	Description
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null
<b>limit</b> integer optional, default is 25	The maximum number of categories to return, up to 50.

### Retrieve Specific Categories

This call returns all available form categories including their respective department.

Fetch a Properties categories as categories Object array.

```
GET v3/categories/:category_id
```

### Request URL Parameters

Parameter	Description
<b>id</b> integer required	The category id
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null

## Requests

### Retrieve All Requests

This call returns all requests for property and their roles. Requests are returned in a paginated collection which provides a Meta Array with pagination information.

Fetch a Properties requests as request Object array.

```
GET v3/requests
```

#### Request URL Parameters

Parameter	Description
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null
<b>limit</b> integer optional, default is 25	The maximum number of requests to return, up to 50.
<b>offset</b> integer optional, default is 0	The number of requests to skip over, where the ordering is consistent but unspecified.
<b>status</b> string optional, default is 0	Filter Requests that match a specific status. supported strings "new, assigned, processing, completed, cancelled, unresolved"  Use Unresolved to retrieve all requests that have any status other than completed or cancelled.
<b>room</b> string optional	Filter Requests that are assigned to a single specific room.
<b>rooms</b> array optional	Filter Requests that belong to an array of rooms. i.e. <code>&amp;rooms[]=32&amp;rooms[]=37&amp;rooms[]=49</code> would return requests that have rooms 32, 37 & 49
<b>category</b> string optional	Category name from the <a href="#">categories endpoint</a>

### Retrieve Single Request

This call returns a specific request for property.

Fetch a Properties request as request Object array.

```
GET v3/requests/:request_id
```

#### Request URL Parameters

Parameter	Description
-----------	-------------

<b>id</b> integer required	The request id
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null

## Create a Request

This call creates a new request and returns a request object.

```
POST v3/requests
```

### Request URL Parameters

Parameter	Description
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null
<b>Guest</b> array of strings required	see list of <a href="#">supported strings</a>
<b>Category</b> string required	form category name from the <a href="#">categories endpoint</a>
<b>room</b> string optional	Guest room for their booking
<b>booking_id</b> string optional	The reservation identifier for the guests stay with the property, if applicable.
<b>guest_notify</b> boolean required	Should an email be sent to the guest notifying them of this requests creation? Recommended false
<b>staff_username</b> string optional	Staff Username from the <a href="#">staff endpoint</a> if provided the newly created request will be assigned to this staffer.
<b>status</b> string optional default "New"	Request status may managed by the api supported strings "new, assigned, processing, completed, cancelled"
<b>form_name</b> string required	Name of the form we are creating this request for from the <a href="#">forms endpoint</a>
<b>form_fields</b> array required	details of the form request. Needs to match any required form_fields for the specified form_name from the <a href="#">forms forms endpoint</a>

### Example Request

```
{
  "hotel_id": 2,
  "room": "32",
  "booking_id": "1332025-4X",
  "guest": {
    "email": "joe@banks.com",
    "name": "Joe Banks",
    "phone": "7075551414"
  },
  "guest_notify": false,
  "form_category": "Treatment",
  "form_name": "Spa Menu",
  "form_fields": {
    "Notes": "We'd like hot towels and a room to meditate in",
    "swedish1": 2,
    "pregnancy2": 1
  }
}
```

### Example Response



```

{
  "requests": {
    "id": 878,
    "booking_id": "1332025-4X",
    "room": "32",
    "guest": {
      "name": "Joe Banks",
      "email": "joe@banks.com",
      "phone_number": "7075551414"
    },
    "created_at": "2020-02-12 15:43:15",
    "updated_at": "2020-02-12 15:43:15",
    "length_open": "1 second",
    "status": "New",
    "assigned": null,
    "category": {
      "id": 8,
      "name": "Treatment"
    },
    "request_details": [
      {
        "name": "Notes",
        "value": "We'd like hot towels and a room to meditate in",
        "amount": null,
        "type": "text"
      },
      {
        "name": "Swedish1",
        "value": "2",
        "amount": "80",
        "type": "amount"
      },
      {
        "name": "Pregnancy2",
        "value": "1",
        "amount": "140",
        "type": "amount"
      }
    ],
    "comments": {
      "guest_comments": [],
      "staff_notes": []
    }
  },
  "api_version": 3,
  "status": "success",
  "request_uuid": "a25a7d79-ba8a-40a3-92a6-02862f7a16c2"
}

```

## Update a Request

This call creates a new request and returns a request object.

```
PUT/PATCH v3/requests/:request_id
```

### Request URL Parameters

Parameter	Description
-----------	-------------



<b>id</b> integer required	The ID of the request resource being managed
<b>hotel_id</b> integer required	Property Numerical identifier Cannot be null
<b>Guest</b> array of strings optional	see list of <a href="#">supported strings</a>
<b>Category</b> string required	form category name from the <a href="#">categories endpoint</a>
<b>room</b> string optional	Guest room for their booking
<b>booking_id</b> string optional	The reservation identifier for the guests stay with the property, if applicable.
<b>guest_notify</b> boolean required	Should an email be sent to the guest notifying them of this requests update? Recommended false
<b>staff_username</b> string optional	Staff Username from the <a href="#">staff endpoint</a> if provided the newly created request will be assigned to this staffer.
<b>status</b> string optional	Request status may managed by the api supported strings "New, Assigned, Processing, Completed, Cancelled"
<b>form_name</b> string required	Name of the form we are creating this request for from the <a href="#">forms endpoint</a>
<b>form_fields</b> array required	details of the form request. Needs to match any required form_fields for the specified form_name from the <a href="#">forms forms endpoint</a>

## Requests Guest Array

Parameter	Description
<b>name</b> string required	Guest fullname i.e. "John Doe"
<b>email</b> string required	guest email
<b>phone</b> string optional	guests preferred phone number

## Webhooks

Properties can enable a webhook notification url at the property level. If enabled we will send a json request object to the notification url every time a request is created or updated. The property has control over if they wish to use a notification url or not, and may turn it on or off by editing their property details as shown below.

Name	A MyPMS Demo Hotel
Organization	BookingCenter
Site URL	http://Hotel.bookingcenter.com
Property Contact Email	info@bookingcenter.com
Timezone	(UTC-04:00) America/New_York - 3:47 PM
Authorized Api Vendors	Wend Your Way  BookingCenter 
<b>External notification webhook</b> <i>If enabled, a json request object will be sent to this url on all request creation and status changes.</i>	
<div>Disable</div>	
Webhook Url	https://yourdomain.com/webhook/index.html
Basic Auth Username	foo
Basic Auth Password	bar

## Requirements

- The webhook url must support HTTPS. We will automatically convert any http or non defined url to https upon saving the webhook url.
- The webhook url must support POST method.
- The webhook url must support BASIC Authentication and the username and password.
- If enabled MyGuests will send a json request object to the webhook url for every request that is created or updated. The partner should use filtering logic as appropriate to capture/discard messages based upon their desired needs.

## Sample Webhook Json Request Object

```
{
  "id": 881,
  "booking_id": "1796110W",
  "room": "32",
  "guest": {
    "name": "Joe Banks",
    "email": "joeb@banks.com",
    "phone_number": null
  },
  "created_at": "2020-02-26 14:48:08",
  "updated_at": "2020-03-10 15:28:34",
  "length_open": "1 week",
  "status": "Completed",
  "assigned": "John Doe - BookingCenter",
  "category": {
    "id": 24,
    "name": "Housekeeping"
  },
  "request_details": [
    {
      "name": "Details",
      "value": "Room Cleaning Required",
      "amount": null,
      "type": "text"
    }
  ],
  "comments": {
    "guest_comments": [
      {
        "details": "Room is being cleaned",
        "created_at": "2020-03-02 12:01:24",
        "staff": "mary"
      },
      {
        "details": "Hi",
        "created_at": "2020-03-02 13:24:54",
        "staff": "tom"
      },
      {
        "details": "Lets Test This out",
        "created_at": "2020-03-02 14:26:02",
        "staff": "ellen"
      }
    ],
    "staff_notes": [
      {
        "details": "Hi",
        "created_at": "2020-03-02 12:01:13",
        "staff": "ellen"
      }
    ]
  }
}
```