**API Specification Doc**

**(Online Hotel Mangement *App*)**

## NOTE: Please don’t request edit access to this demo document. Instead, click [here](https://docs.google.com/document/d/1HSQ3Fe77hnthw8hizqvXJU-qGEPHavMkctvCCadkVbY/template/preview?usp=drive_web) to make a copy of this template for your use. Thank you!

## 

## 

## 

Index

[1. login](#_bit7idv2ancx)

[Request](#_kd6tjl5w5wjo)

[Response](#_8b7ijrpuamb8)

[2. get updates](#_9t094e45sg90)

[Request](#_38pypefkb82)

[Response](#_47awg09ehquu)

[3. deletions](#_9mc75gzfq3m3)

[Request](#_xtszbus9yomc)

[Response](#_xuhh3y4ufz7)

[4. get recipe image](#_p8sbwmpv82e8)

[Request](#_iaps2dh7l2m2)

[Response](#_fvf6lulvf6wv)

[Conventions](#_uk42zio712fb)

[Status Codes](#_of6ke88xwee4)

## 

## 

## 

Methods

## 1. login

Authenticate the user with the system and obtain the auth\_token

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/login/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST  POST | api\_key  username  password | String  string  string |

**api\_key**

api\_key must be sent with all client requests. The api\_key helps the server to validate the request source.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key (**string**) - all further API calls must have this key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"Please provide username."} |
| 400 | {"error":"Please provide password."} |
| 401 | {"error":"Invalid API key."} |
| 401 | {"error":"Incorrect username or password."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 2. get dashboard

Get the Occupancy and room availability

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/dashboard/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  Post | auth\_key  Occupancy  {  “From “:date,  “To”: “date”  },  Room Availability  {  “From date”: “Date”,  “To”: “date”  } | String  object  Object |

**auth\_key**

The auth\_key that was given in response to /api/login

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  “data”:[  "occupied Room": 10,  “Room availability” :[  “RoomID”:int,  “Room Type”:”String”,  “Room availability”:int  ]  ]  } |
| 400 | {"error":"Please missing parameters."} |
| 401 | {"error":"Invalid API key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 3. Edit user

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **PUT** | api/Edituser/<user\_id> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| HEAD  POST | auth\_key  {  First name:string,  Last name:string,  Mobile:int,  Email:string,  DOB:date,  Id/Passport:int  Address:string  City:string,  Zip/Postal:int,  Country:string  Gender:string,  User Role:string  } | String  object |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **user is updated** **Successfully** |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 401 | {"error":"Invalid Auth key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 4. Add Room

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/room/addedit |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | {  Room name:string,  Room type:string,  Floor:int,  Notes(optional):string  } | object |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | Room Created Successfully |
| 500 | {"esrror":"Something went wrong. Please try again later."} |

## 4. Add Room Types

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/roomtypes |
| **PUT** | Api/roomtypes/<Room\_Type> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | {  Room\_Type:string,  Room\_price:string,  Room\_Abbreviation: string,  Maxnoof\_guests:int  } | object |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | RoomType Added Successfully |
| 500 | {"esrror":"Something went wrong. Please try again later."} |

## 6. Add Guest

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/guest |
| **PUT** | api/guest/<GuestID> |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | {  GuestId:string,  First name:string,  Last name:string,  Mobile:int,  Email:string,  DOB:date,  Id/Passport:int  Address:string  City:string,  Zip/Postal:int,  Country:string  Gender:string  } | object |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| **200** | guest Created Successfully |
| 200 | guest is upadted Successfully |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 

Glossary

## Conventions

* **Client** - Client application.
* **Status** - HTTP status code of response.
* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All response are in JSON format.
* All request parameters are mandatory unless explicitly marked as [optional]
* The type of values accepted for a *request* parameter are shown the the values column like this [**10**|<any number>] .The | symbol means *OR*. If the parameter is [optional], the default value is shown in blue bold text, as **10** is written in [**10**|<any number>].

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |