API Documentation

The roomcaptain API can be used to retrieve data concerning the rooms, availability and action bookings. The API can be accessed by a URL, where the search criteria is sent via a form upload and the result is a JSON data sctructure.

There are 5 type of action that the api can handle:

1. Getting list of rooms – this action retrieves the list of rooms available for a location
2. Availability – this action retrieves the rooms and their availability for a certain period
3. Sitetext – this action retrieves the list of keys and their associated value for a specified language
4. Book – this action books a reservation
5. Room calendar – this action retrieves availability of a room for a certain period of time

The api can be access via the following URLs:

- QA http://dev.recepcio.roomcaptain.com/api.php
- Prod http://recepcio.roomcaptain.com/api.php

Errors

When the server cannot handle a request an error will be generated and returned to the caller. If any parameter is missing or if the action parameter is not one of the valid values, a simple text response (in English) will return.

For other errors the JSON response will contain one element with name 'error' and one of the below values:

Error Codes	Explanation
BOOKING_DATE_MUST_BE_IN_THE_FUTURE	The booking date must be in the future
CHECKOUT_DATE_MUST_BE_AFTER_CHECKIN_DATE	The end date has to be after the start date
FOR_SELECTED_DATE_MIN_STAY X	The minimum number of days for the period is X days
FOR_SELECTED_DATE_MAX_STAY X	The maximum number of days for the period is X days
DB_ERROR	DB Error occurred. It worth to retry the request again.

Actions

1. Getting list of rooms

http://dev.recepcio.roomcaptain.com/api.php?lang=eng&location=teszt_hostel&action=rooms

Incoming Parameters

- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'rooms' (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you

Returned data

The returned data is a JSON structure:

```
{
  Room type id: {
    id: the id this is used to addess the room when sending in the booking,
    price_per_bed: this is the preset price of the bed if the room is a DORM,
                (for certain dates this value may change, you will see in the availability request)
    price_per_room:,this is the preset price of the room if the room is a PRIVATE or APARTMENT room
                 (for certain dates this value may change, you will see in the availability request)
    surcharge_per_bed: for apartments the price_per_room is for 2 people. For any additional guest this surcharge is counted,
    type": the type of the room, can be DORM, PRIVAATE, APARTMENT,
    num_of_beds": the number of available beds in the room,
    name": the name of the room,
    description: description of the room,
    short_description: short description of the room,
    size: description of the room relating to its size,
    location: description of the room relating to its location,
    bathroom: description of the room relating to its bathroom,
    _order: ordering of each room on the admin interface. UI may choose to sort the rooms in different order,
    num_of_beds_avail:This value will always be 0 on this query do not use it here,
    images: [
      {
        id: id of the image,
        filename: the file name,
        room_type_id: the id of the room type that this image is linked to,
        width: with of the image in pixels,
        height: height of the image in pixels,
        default: 1 if this is the default image for this room, 0 if it is not,
        _order: the order of the image set on the admin inter,
        thumb: the name of the thumb image file,
        medium: the name of the medium image file,
        original_img_url: URL of the original image. Use this directly to show the image on a page,
        medium_img_url: URL of the medium image. Use this directly to show the image on a page,
        thumb_img_url: URL of the thumb image. Use this directly to show the image on a page
      },
      …
    ]
}
```

2. Getting the availability

http://dev.recepcio.roomcaptain.com/api.php?lang=eng&location=teszt_hostel&action=availability&from=2017-01-05&to=2017-01-07&filter_room_types=42,35

Incoming Parameters

- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'availability (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you
- from   date of the arrival (format: YYYY-MM-DD)
- to   date of the departure (format: YYYY-MM-DD)
- filter_room_types (optional parameter) coma separated list of room type ids that should return only.
                          If this parameter is omitted then all room types will return

Returned data

The returned data is a JSON structure:

```
{
  "special_offers": {
    <special offer id>: {
      id: id of the special offer,
      name: name of the special offer (not for display to users),
      start_date: start date of the special offer. Use rather the dates array below where multiple ranges can be set,
      end_date: end date of the special offer. Use rather the dates array below where multiple ranges can be set,
      discount_pct: discount percentage (if this value is 10 it means that the price will be 10% less than normal if this applies,
      nights": the minimum number of nights that this special offer applies,
      valid_num_of_days_before_arrival": arrival date must be at least as many days in the future as this number
      visible": 1 is visible 0 is not visible,
      room_type_ids: comma separated list of room type ids where this special offer applies,
      early_bird_day_count": arrival date must be at most as many days in the future as this number,
      title: title of the special offer,
      text": description of the special offer,
      room_name": the name of the room or rooms that this special offer applies to (like all private rooms ,etc),
      dates": [
        {
          "start_date": start of the validity of the special offer (format: YYYY-MM-DD),
          "end_date": end of the validity of the special offer (format: YYYY-MM-DD)
        },
        …
      ]
    },
    …
  },
  "rooms": [
    {
      id: the id this is used to addess the room when sending in the booking,
      price_per_bed: this is the preset price of the bed if the room is a DORM,
                  (for certain dates this value may change, you will see in the availability request)
      price_per_room:,this is the preset price of the room if the room is a PRIVATE or APARTMENT room
                  (for certain dates this value may change, you will see in the availability request)
      surcharge_per_bed: for apartments the price_per_room is for 2 people. For any additional guest this surcharge is counted,
      type": the type of the room, can be DORM, PRIVAATE, APARTMENT,
      num_of_beds": the number of available beds in the room,
      name": the name of the room,
      description: description of the room,
      short_description: short description of the room,
      size: description of the room relating to its size,
      location: description of the room relating to its location,
      bathroom: description of the room relating to its bathroom,
      _order: ordering of each room on the admin interface. UI may choose to sort the rooms in different order,
      num_of_beds_avail:Applicable for DORMs, this is the number of beds available for the time period
      num_of_rooms_avail: Applicable for PRIVATE and APARTMENTs, this is the number of rooms available for the time period,
      price: This is the price of bed (for DORM) or price of room (for PRIVATE) per day for the period.
            This price is the average price for the period so if you book for 2 days, with 10 and 12 the price for the 2 days
            this value will be 11.
            This price also includes any discounts that are applicable if a special offer applies.
            So you do not need to calculate anything related to the price, just show this field as amount/day/(bed or room)
      price_2: For apartment the price if it is booked for 2 people,
      price_3: For apartment the price if it is booked for 3 people,
      price_4: For apartment the price if it is booked for 4 people,
      price_5: For apartment the price if it is booked for 5 people,
      price_6: For apartment the price if it is booked for 6 people,
      special_offer: the special offer that applies to this room for the period of the booking,
      special_offer_for_one_more_day: the special offer that would apply to this room for the period of the booking if the guest
                                  stayed an extra day,
      images: [
        {
          id: id of the image,
          filename: the file name,
          room_type_id: the id of the room type that this image is linked to,
          width: with of the image in pixels,
          height: height of the image in pixels,
          default: 1 if this is the default image for this room, 0 if it is not,
          _order: the order of the image set on the admin inter,
          thumb: the name of the thumb image file,
          medium: the name of the medium image file,
          original_img_url: URL of the original image. Use this directly to show the image on a page,
          medium_img_url: URL of the medium image. Use this directly to show the image on a page,
          thumb_img_url: URL of the thumb image. Use this directly to show the image on a page
        },
        …
      ]
    },
    …
  ]
}
```

3. Getting the texts for the website

Incoming Parameters

- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'dictionary' (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you

Returned data

The returned data is a JSON structure:

```
{
  Key1: value1,
  Key2: value2,
  …
}
```

The key value pairs can be set in http://dev.recepcio.roomcaptain.com/view_sitetexts.php (for the prod version remove the dev from the beginning of the URL)

4. Getting a list of services that can be booked along with the reservation

Incoming Parameters

- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'services' (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you

Returned data

The returned data is a JSON structure:

```
[
  {
    id: The ID of the service. This will need to be specified when sending in the booking if the service is selected,
    price: The price of the service,
    currency: The currency the price is in,
    free_service: 0 if it's a free service, 1 if it's a paying service,
    title: The title of the service,
    description: The description of the service,
    unit_name: The unit, for example the taxi from the airport is "4 passenger", so when you display it it will be price/unit
  },
  …
]
```

5. Book a reservation

Incoming Parameters

- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'book' (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you
- firstname The first name of the guest
- lastname   The last name of the guest
- email   The email of the guest
- phone   The phone of the guest
- nationality The nationality of the guest (use the 2 character code

- street   The street address of the guest
- city   The city of the guest
- zip   The zip of the guest
- country   The country of the guest
- comment   The comment of the reservation
- booking_data   The booking data, this is a json structure containing the rooms/beds to book. See below.
- from_date   The arrival date (format: YYYY-MM-DD)
- to_date   The departure date (format: YYYY-MM-DD)
- services   The selected services for the reservation. It is a JSON data structure. See below.

The **booking_data** has the following json structure:

```
[
  {
     roomTypeIds: array of room type ids, this will normally be an array of 1 element,
     numOfPerson: array of number where each, element is the number of person into 1 room
  },
  …
]
```

Examples:

2 private room booked, 1 double bed room (room_type_id: 42) and 1 four bed room (room_type_id 43):
```
[
  { roomTypeIds: [42], numOfPerson:[2]},
  { roomTypeIds: [43], numOfPerson:[4]}
]
```

2 private room booked, both are double bed room (room_type_id: 42)
```
[
  { roomTypeIds: [42], numOfPerson:[2,2]}
]
```
or
```
[
  { roomTypeIds: [42], numOfPerson:[4]}
]
```
or
```
[
  { roomTypeIds: [42], numOfPerson:[2]},
  { roomTypeIds: [42], numOfPerson:[2]}
]
```

1 private double room booked (room_type_id: 42) and 4 bed from a dorm (room_type_id: 51)
```
[
  { roomTypeIds: [42], numOfPerson:[2]},
  { roomTypeIds: [51], numOfPerson:[4]}
]
```

2 apartment is booked (room_type_id: 62), 1 with 2 guest the other one with 3 guests
```
[
  { roomTypeIds: [61], numOfPerson:[2,3]},
]
```
or
```
[
  { roomTypeIds: [61], numOfPerson:[2]},
  { roomTypeIds: [61], numOfPerson:[3]},
]
```

1 private room booked, either a double room (room_type_id: 42) or a triple room (room_type_id: 43)
```
[
```

```
  { roomTypeIds: [42,43], numOfPerson:[2]}
]
```

The **services** has the following json structure:

```
[
  {
      serviceId: the ID of the service. This is acquired from the loadServices action,
      occasion: the number of times the services wants to be used. For example for a bike rental the number of bikes to rent,
      comment: any additional comment with regards to the reservation of this service (for example the day when to use it)
  },
  …
]
```

Returned data
There is no returned data for this action

6. *Room Calendar*

Incoming Parameters
- lang   The 3 character language identifier (currently supported are: deu, eng, fra, ita, esp)
- action   'room_avalability' (without the quotes)
- location   For dev use 'teszt_hostel'. For prod use the name that is provided to you
- room_type_id   The id of the room type that we want to get the availability
- from   Start date from where we want the availability
- to   End date until we want the availability

Returned data
The returned data is a JSON structure:

```
[
  {
    date: the date for which the availability is given (format: YYYY-MM-DD),
    numberOfAvailableBeds: The number of available beds. Should be used for DORM types only,
    numberOfAvailableRooms: The number of available rooms. Should be used for PRIVATE or APARTMENT rooms only.
                            For DORM this will be 0 always
  },
  …
]
```