

# System obsługi rezerwacji małego hotelu

## Dokumentacja API

Wersja 0.3.0

Opracował: Michał Wojczewski

## Spis treści

<b>1.</b>	<b>Wprowadzenie .....</b>	<b>3</b>
<b>2.</b>	<b>Podstawy połączenia .....</b>	<b>3</b>
<b>3.</b>	<b>Dostępne opcje.....</b>	<b>4</b>
<b>4.</b>	<b>Opis szczegółowy opcji.....</b>	<b>5</b>
a.	<i>Lista pokoi .....</i>	<i>5</i>
b.	<i>Szczegółowy opis pokoju .....</i>	<i>6</i>
c.	<i>Zdjęcia pokoju .....</i>	<i>7</i>
d.	<i>Lista rezerwacji.....</i>	<i>8</i>
e.	<i>Kontrola rezerwacji .....</i>	<i>9</i>
f.	<i>Tworzenie nowej rezerwacji.....</i>	<i>10</i>
<b>5.</b>	<b>Panel administracyjny.....</b>	<b>12</b>
<b>6.</b>	<b>Historia zmian.....</b>	<b>12</b>

## 1. Wprowadzenie

Opisywany system zapewnia wymianę danych pomiędzy backendem serwerowym (PHP z frameworkiem Symfony) a zewnętrznym klientem, obsługującym frontend.

Dane dostępne za pośrednictwem API:

- a. Lista wszystkich dostępnych pokoi w hotelu
- b. Szczegółowy opis wybranego pokoju
- c. Lista zdjęć połączonych z wybranym pokojem
- d. Lista rezerwacji połączonych z danym pokojem
- e. System weryfikacji dostępności pokoju do rezerwacji w wybranym okresie

## 2. Podstawy połączenia

System dostępny jest pod poniższym adresem:

```
http://hotel.wojczewski.pl
```

Całość wymiany informacji oparta jest o system API:

- a. GET – pobranie informacji (READ)
- b. POST – wysłanie nowych informacji (CREATE)
- c. PUT – nadpisanie informacji (UPDATE)
- d. DELETE – kasowanie informacji (DELETE)

Odpowiedź na żądanie generowana jest w postaci JSON.

Polityka CORS nie została ograniczona na etapie projektowania systemu. Końcowa wersja będzie ograniczać odpowiedź tylko do wywołań z własnej domeny.

### 3. Dostępne opcje

Lista pokoi	GET /api/rooms
Szczegółowy opis pokoju	GET /api/room/details
Zdjęcia pokoju	GET /api/room/photos
Lista rezerwacji	GET /api/room/reservations
Kontrola rezerwacji	GET /api/room/reservations/check
Tworzenie nowej rezerwacji	POST /api/rooms/reservations/create

## 4. Opis szczegółowy opcji

### a. Lista pokoi

Wywołanie:

```
GET /api/rooms
```

Odpowiedź (tablica obiektów):

```
[
  {
    "id":1,
    "name":"Zielony",
    "room_no":1,
    "floor_no":1,
    "size":50,
    "beds":5,
    "balcony":false,
    "breakfast":true,
    "pets_allowed":true,
    "price":250
  },
  {...}
]
```

**id** – (*integer*) identyfikator pokoju - na jego podstawie udostępniane są pozostałe dane

**name** – (*string*) nazwa pokoju

**room\_no** – (*integer*) numer pokoju

**floor\_no** – (*integer*) piętro, na którym znajduje się pokój

**size** – (*integer*) wielkość pokoju (wyrażana w m<sup>2</sup>)

**beds** – (*integer*) ilość łóżek w pokoju (definiuje ilość gości)

**balcony** – (*boolean*) czy pokój posiada balkon/taras?

**breakfast** – (*boolean*) czy w cenie pokoju zawarte jest śniadanie?

**pets\_allowed** – (*boolean*) czy w pokoju można posiadać zwierzęta?

**price** – (*float*) cena za dobę hotelową

## b. Szczegółowy opis pokoju

Wywołanie:

```
GET /api/room/details/<id>
```

Parametry wywołania:

**<id>** - (integer) identyfikator pokoju

Odpowiedź (obiekt):

```
{
  "id":1,
  "name":"Zielony",
  "description": "Przykładowy opis pokoju."
  "room_no":1,
  "floor_no":1,
  "size":50,
  "beds":5,
  "balcony":false,
  "breakfast":true,
  "pets_allowed":true,
  "price":250,
  "roomPhotos":
  [
    {
      "id":1,
      "name":"72a158d582967697f0f0f78470bd6179.jpg"
    }
  ]
}
```

**id** – (integer) identyfikator pokoju - na jego podstawie udostępniane są pozostałe dane

**name** – (string) nazwa pokoju

**description** – (string) opis pokoju

**room\_no** – (integer) numer pokoju

**floor\_no** – (integer) piętro, na którym znajduje się pokój

**size** – (integer) wielkość pokoju (wyrażana w m<sup>2</sup>)

**beds** – (integer) ilość łóżek w pokoju (definiuje ilość gości)

**balcony** – (boolean) czy pokój posiada balkon/taras?

**breakfast** – (boolean) czy w cenie pokoju zawarte jest śniadanie?

**pets\_allowed** – (boolean) czy w pokoju można posiadać zwierzęta?

**price** – (float) cena za dobę hotelową

**roomPhotos** – (array) tablica obiektów z nazwami zdjęć przypisanymi do pokoju:

**id** – (integer) identyfikator zdjęcia

**name** – (string) nazwa pliku zdjęcia

Zdjęcia znajdują się w katalogu **/uploads/images/**

c. Zdjęcia pokoju

Wywołanie:

```
GET /api/room/photos/<id>
```

Parametry wywołania:

<id> - (integer) identyfikator pokoju

Odpowiedź (tablica obiektów):

```
{
  "roomPhotos":
  [
    {
      "id": 1,
      "name": "72a158d582967697f0f0f78470bd6179.jpg"
    }
  ]
}
```

**id** – (integer) identyfikator zdjęcia

**name** – (string) nazwa pliku zdjęcia

Zdjęcia znajdują się w katalogu **/uploads/images/**

#### d. Lista rezerwacji

Wywołanie:

```
GET /api/room/reservations/<id>
```

Parametry wywołania:

<id> - (integer) identyfikator pokoju

Odpowiedź (tablica obiektów):

```
{
  "reservations":
  [
    {
      "id":1,
      "created_at":"2022-11-18T00:00:00+00:00",
      "starts_at":"2022-12-21T13:00:00+00:00",
      "ends_at":"2022-12-27T11:00:00+00:00",
      "paid":true
    },
    {...}
  ]
}
```

**id** - (integer) identyfikator rezerwacji

**created\_at** - (string) data utworzenia rejestracji

**starts\_at** – (string) data początkowa rezerwacji

**ends\_at** – (string) data końcowa rezerwacji

**paid** – (Boolean) czy rezerwacja została opłacona?



#### e. Kontrola rezerwacji

Wywołanie:

```
GET /api/room/reservations/check/<id>/<yyyy>/<mm>/<dd>/<yyyy>/<mm>/<dd>
```

Parametry wywołania:

**<id>** - (integer) identyfikator pokoju

**<yyyy>** - (integer) rok początku rezerwacji

**<mm>** - (integer) miesiąc początku rezerwacji

**<dd>** - (integer) dzień początku rezerwacji

**<yyyy>** - (integer) rok końca rezerwacji

**<mm>** - (integer) miesiąc końca rezerwacji

**<dd>** - (integer) dzień końca rezerwacji

Odpowiedź (obiekt):

```
{  
  "conflicts": 1  
}
```

**conflicts** – (integer) liczba rezerwacji kolidujących z testowaną

Z założenia, dokonanie nowej rezerwacji powinno odbyć się po zweryfikowaniu, czy termin nie koliduje z już istniejącymi rezerwacjami. Powyższe zapytanie powinno zwrócić zatem wynik 0.

f. Tworzenie nowej rezerwacji

Wywołanie:

```
POST /api/rooms/reservations/create
```

Przykład zapytania dodającego nową rezerwację:

```
fetch('http://hotel.wojczewski.pl/api/room/reservations/create', {  
  method: 'POST',  
  body: JSON.stringify({  
    room_id: <id>,  
    starts_at: "<yyyy>.<mm>.<dd>",  
    ends_at: "<yyyy>.<mm>.<dd>",  
    paid: <opłacone>  
  }),  
  headers: {  
    'Content-type': 'application/json; charset=UTF-8'  
  }  
})
```

<id> - (integer) identyfikator pokoju

<yyyy> - (integer) rok początku rezerwacji

<mm> - (integer) miesiąc początku rezerwacji

<dd> - (integer) dzień początku rezerwacji

<yyyy> - (integer) rok końca rezerwacji

<mm> - (integer) miesiąc końca rezerwacji

<dd> - (integer) dzień końca rezerwacji

<opłacone> - (boolean) czy rezerwacja została opłacona? (0 – nie, 1 – tak)

Możliwe odpowiedzi serwera:

- błędy:

```
{
  "error": "data_not_found"
}
```

*#serwer nie znalazł danych w zapytaniu.*

```
{
  "error": "room_not_found"
}
```

*#serwer nie znalazł określonego pokoju.*

```
{
  "error": "range_overlap"
}
```

*#istnieje konflikt terminów dla rezerwacji.*

- rezerwacja dodana poprawnie:

```
{
  "id": 4,
  "created_at": "2022-11-26T17:48:51+00:00",
  "starts_at": "2022-02-01T13:00:00+00:00",
  "ends_at": "2022-03-03T11:00:00+00:00",
  "paid": true
}
```

## 5. Panel administracyjny

W celu uproszczenia prac związanych z uruchomieniem projektu, został wytworzony tymczasowy panel administracyjny, pozwalający na edycję podstawowych danych (pokoje, zdjęcia, rezerwacje)

Dostęp do panelu możliwy jest, po zalogowaniu, pod adresem:

<http://hotel.wojczewski.pl/admin>

## 6. Historia zmian

Data edycji	Wprowadzone zmiany
22.11.2022	V 0.1.0 Wersja wstępna
22.11.2022	V 0.1.1 Zmiana odpowiedzi dla testowania konfliktów rezerwacji
24.11.2022	V 0.2.0 Dodanie pola description do pokoju Wygenerowanie panelu administracyjnego
26.11.2022	V 0.3.0 Opcja dodawania nowej rejestracji poprzez API