**API serwera IDC Software Hotel Suite**

**JSON**

**wersja 0.35**

w trakcie tworzenia

Serwer IDC Software Suite jest serwerem WebSocket. Do jego działania wymagane jest środowisko Node JS z zainstalowanymi pakietami ws i mysql. Jako, że jest to aplikacja dostępowa do bazy danych, wymagana jest też działająca baza danych MySql, załączona do aplikacji. Komunikacja odbywa się za pomocą tzw. pakietów opisanych w języku JSON.

Każdy pakiet posiada pole command, które determinuje pośrednio tablicę, na której chce operować użytkownik. Niektóre pakiety mają pole action, które pozwala na standardowe operacje CRUD na wskazanych polem command danych.

**Login**

Pola:

command – "login"

loginData.userEmail – string

loginData.userPasswordHash - string

Przykładowy pakiet:

{

"command": "login",

"loginData":{

"userEmail": "[a@a.com](mailto:a@a.com)",

"userPasswordHash": "abcd"

}

}

Pakiet zaloguje użytkownika na konto o podanych danych.

**Register**

Pola:

command – "register"

registerData.userId – string

registerData.userPermissionLevel – string

registerData.userFirstName – string

registerData.userSecondName – string

registerData.userLastName – string

registerData.userEmail – string

registerData.userPasswordHash – string

registerData.userHotelId - int

Przykładowy pakiet:

{

"command": "register",

"registerData":{

"userId": "B12345",

"userPermissionLevel": "guest",

"userFirstName": "Jan",

"userSecondName": "",

"userLastName": "Kowalski",

"userEmail": "[v@a.com](mailto:v@a.com)",

"userPasswordHash": "zxcasd",

"userHotelId": null

}

}

Pakiet zarejestruje nowego użytkownika. Pole userHotelId wskazuje na jego charakter, jeśli jest null – jest to gość, jeśli nie – jest to pracownik hotelu o podanym id.

**UserPermissionLevel**

Pakiet pozwala dokonywać operacji na poziomach uprawnień, jakie można przydzielić użytkownikowi.

Pola:

command – "permissionLevel"

action - string

UserPermissionsLevelName – string, identyfikator

ManageHotels - bool

ManageRooms - bool

ManageGuests – bool

ManageEmployees – bool

ManageReservations – bool

Akcje:

*get* – zwraca listę dostępnych poziomów uprawnień. Pole z listą ma nazwę "list", a poszczególne pola wierszy są tożsame z polami w bazie danych (i tym samym z tymi w tym pakiecie poza command i action).

*add* – dodaje nowy poziom uprawnień. Wszystkie pola muszą być sprecyzowane.

*update* – uaktualnia istniejący już poziom uprawnień. Nie można zmienić jego nazwy, w takim wypadku należy usunąć stary poziom i dodać nowy, poprawny. Wszystkie pola muszą być sprecyzowane.

*delete* – usuwa istniejący poziom uprawnień. Wszystkie pola poza nazwą nie są wymagane i zostaną zignorowane.

Przykładowy pakiet:

{

"command": "permissionLevel",

"action": "update",

"UserPermissionLevelName": "superadmin",

"ManageHotels": true,

"ManageRooms": true,

"ManageGuests": true,

"ManageEmployees": true,

"ManageReservations": true

}

Pakiet uaktualni poziom uprawnień "superadmin" w taki sposób, że ustawi uprawnienia na żądane wartości.

**Hotel**

Pakiet pozwala operować na hotelach w bazie danych.

Pola:

command – "hotel"

action - string

HotelId – liczba (autoincrement)

HotelName – string

HotelCountry – string

HotelCity – string

HotelStreet – string

HotelRating – int

HotelEmail – string

HotelPhone - string

Akcje:

*get* – zwraca listę dostępnych hoteli. Pole z listą ma nazwę "list", a poszczególne pola wierszy są tożsame z polami w bazie danych (i tym samym z tymi w tym pakiecie poza command i action).

*add* – dodaje nowy hotel. Wszystkie pola muszą być sprecyzowane poza HotelId, które zostanie zignorowane.

*update* – uaktualnia istniejący już hotel. Nie można zmienić jego id, w takim wypadku należy usunąć stary poziom i dodać nowy, poprawny. Wszystkie pola muszą być sprecyzowane.

*delete* – usuwa istniejący hotel. Wszystkie pola poza HotelId nie są wymagane i zostaną zignorowane.

Przykładowy pakiet:

{

"command": "hotel",

"action": "add",

"HotelName": "Fajny hotel",

"HotelCountry": "pl",

"HotelCity": "Koszalin",

"HotelStreet": "Majowa 5",

"HotelRating": 5,

"HotelEmail": [fajny@hotel.pl](mailto:fajny@hotel.pl),

"HotelPhone": "555-444-333"

}

Pakiet doda nowy hotel o podanych danych.

**Room**

Pakiet pozwala operować na pokojach w bazie danych.

Pola:

command – "room"

action - string

HotelId – int

RoomNumber – int

TemplateId - string

Akcje:

*get* – zwraca listę dostępnych pokoi we wszystkich hotelach (można po stronie klienta rozróżniać po HotelId). Pole z listą ma nazwę "list", a poszczególne pola wierszy są tożsame z polami w bazie danych (i tym samym z tymi w tym pakiecie poza command i action).

*add* – dodaje nowy pokój. Wszystkie pola muszą być sprecyzowane. Warto zauważyć, że HotelId i TemplateId powinny być wybierane z ComboBoxów, bo zależą od innych tabel.

*update* – uaktualnia istniejący już pokój. Nie można zmienić jego hotelu ani numeru, w takim wypadku należy usunąć stary poziom i dodać nowy, poprawny. Wszystkie pola muszą być sprecyzowane.

*delete* – usuwa istniejący pokój. Wymagane są tylko pola HotelId i RoomNumber.

Przykładowy pakiet:

{

"command": "room",

"action": "delete",

"HotelId": 1,

"RoomNumber": 201

}

Pakiet usunie pokój o numerze 201 z hotelu o id 1.

**Reservation**

Pakiet pozwala dokonywać zmian w tabeli z rezerwacjami.

Pola:

command – "reservation"

action - string

ReservationId – int (autoincrement)

HotelId – int

RoomNumber – int

UserId – string

ReservationCheckIn – string postaci "YYYY-MM-DD"

ReservationCheckOut – string postaci "YYYY-MM-DD"

Akcje:

*get* – zwraca listę wszystkich rezerwacji. Pole z listą ma nazwę "list", a poszczególne pola wierszy są tożsame z polami w bazie danych (i tym samym z tymi w tym pakiecie poza command i action).

*add* – dodaje nową rezerwację. Wszystkie pola muszą być sprecyzowane. Pola HotelId, RoomNumber i UserId powinny być w postaci ComboBoxów. Jeśli UserId nie istnieje, należy go wcześniej dodać do tablicy z uprawnieniami gościa (w tle podając tylko numer jego dowodu).

*update* – uaktualnia istniejącą już rezerwację. Nie można zmienić jej id, w takim wypadku należy usunąć starą i dodać nową. Wszystkie pola muszą być sprecyzowane.

*delete* – usuwa istniejącą już rezerwację. Wszystkie pola poza ReservationId nie są wymagane i zostaną zignorowane.

Przykładowy pakiet:

{

"command": "reservation",

"action": "get"

}

Pakiet zwróci całą historię rezerwacji.

**Template**

Pakiet pozwala operować na szablonach pokoi.

Pola:

command – "template"

action – string

TemplateId – string

RoomTemplateName – string

RoomTemplateCost – float

RoomTemplateDescription – string

Akcje:

*get* – zwraca listę wszystkich szablonów pokoi. Pole z listą ma nazwę "list", a poszczególne pola wierszy są tożsame z polami w bazie danych (i tym samym z tymi w tym pakiecie poza command i action).

*add* – dodaje nowy szablon pokoju. Wszystkie pola muszą być sprecyzowane.

*update* – uaktualnia istniejący już szablon pokoju. Nie można zmienić jego TemplateId, w takim wypadku należy usunąć stary i dodać nowy. Wszystkie pola muszą być sprecyzowane.

*delete* – usuwa istniejący szablon. Wszystkie pola poza TemplateId nie są wymagane i zostaną zignorowane.

Przykładowy pakiet:

{

"command": "template",

"action": "add",

"TemplateId": "lux1",

"RoomTemplateName": "Apartament luksusowy 1",

"RoomTemplateCost": 119,99,

"RoomTemplateDescription": "blablabla"

}

Pakiet doda nowy szablon pokoju.