Rent-api dokumentacija

# Global

## Output

Podatki so v json obliki(custom format). V “data” arrayu se nahajajo podatki iz baze. V “meta” arrayu se nahaja “count”, ki ima podatek Koliko vrstic je izpisanih/effectanih. V “links” arrayu se nahaja “endpoint”, ki ima podatek na kateremu endpointu je bil API klican.

## Data format

Podatki iz baze se delijo z znakom “+”(primer: {ime+priimek+email+username})

## Host

Dostop do API-ja je na localhostu s portom 8080 z dodanim endpointom in optional parametri (primer: “localhost:8080/users/getSpec?username=test”)

# Users

## GetAll

**Endpoint**: /users

**Format**: “{firstname}+{surname}+{email}+{username}+{password}”

**Opis**: Izpiše vse userje iz baze.

## GetSpecific

**Endpoint**: /users/getSpec

**Format**: “{firstname}+{surname}+{email}+{username}+{password}”

**Parametri**: {username: String}

**Opis**: Izpiše podatke o določenim userju

## UpdateUser

**Endpoint**: /users/updateUser

**Parametri**: {old\_username: String, new\_username: String, new\_firstname: String, new\_surname: String, new\_password: String }

**Opis**: Posodobi userja, ki ima name = old\_username

## InsertUser

**Endpoint**: /users/insertUser

**Parametri**: {new\_firstname: String, new\_surname: String, new\_birth: String, new\_city\_id: int, new\_email: String, new\_username: String, new\_password: String }

**Opis**: V bazo vpiše novega userja

## DeleteUser

**Endpoint:** /users/deleteUser

**Parametri:** {user\_id: int}

**Opis:** iz baze izbriše userja s podanim id-jem

## ResetPassword

**Endpoint:** /users/resetPassword

**Parametri:** {username: String, password: String }

**Opis:** ponastavi geslo podanim userju

# Cars

## GetAll

**Endpoint**: /cars

**Format**: “{name}+{licenceplate}+{garage\_id}+{kilometers}”

**Opis**: Izpiše vse avtomobile iz baze.

## GetSpecific

**Endpoint**: /cars/getSpec

**Format**: “{name}+{licenceplate}+{garage\_id}+{kilometers}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določenim avtomobilu

## UpdateCar

**Endpoint**: /cars/updateCar

**Parametri**: {car\_id: int, new\_name: String, new\_licenceplate: String, new\_garage\_id: int, new\_kilometers: int}

**Opis**: Posodobi avtomobil, ki ima id = car\_id

## InsertCar

**Endpoint**: /cars/insertCar

**Parametri**: { new\_name: String, new\_licenceplate: String, new\_garage\_id: int, new\_kilometers: int }

**Opis**: V bazo vpiše nov avtomobil

## DeleteUser

**Endpoint:** /cars/deleteCar

**Parametri:** {car\_id: int}

**Opis:** iz baze izbriše avtomobil s podanim id-jem

# Cities

## GetAll

**Endpoint**: /cities

**Format**: “{name}+{post\_number}”

**Opis**: Izpiše vse kraje iz baze.

## GetSpecific

**Endpoint**: /cities/getSpec

**Format**: “{name}+{post\_number}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določenem kraju

## UpdateCity

**Endpoint**: /cities/updateCity

**Parametri**: { old\_name: String, new\_name: String, new\_post: int}

**Opis**: Posodobi kraj, ki ima name = old\_name

## InsertCity

**Endpoint**: /cities/insertCity

**Parametri**: { new\_name: String, new\_post: int}

**Opis**: V bazo vpiše nov kraj

## DeleteCity

**Endpoint:** /cities/deleteCity

**Parametri:** {city\_id int}

**Opis:** iz baze izbriše kraj s podanim id-jem

# Confirmed Rents

## GetAll

**Endpoint**: /confirmedrents

**Format**: “{car\_id}+{kilometersmade}+{comment}+{rent\_id}”

**Opis**: Izpiše vse potrjene izposoje iz baze.

## GetSpecific

**Endpoint**: / confirmedrents /getSpec

**Format**: “{car\_id}+{kilometersmade}+{comment}+{rent\_id}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določeni potrjeni izposoji

## UpdateConfirmedRent

**Endpoint**: / confirmedrents /updateConfirmedRent

**Parametri**: {new\_id: int, new\_car\_id: int, new\_kilometersmade: int, new\_comment: String, new\_rent\_id: int}

**Opis**: Posodobi izposojo, ki ima id = new\_id

## InsertConfirmedRent

**Endpoint**: / confirmedrents /insertConfirmedRent

**Parametri**: { new\_car\_id: int, new\_kilometersmade: int, new\_comment: String, new\_rent\_id: int}

**Opis**: V bazo vpiše novo izposojo

## DeleteConfirmedRent

**Endpoint:** / confirmedrents /deleteConfirmedRent

**Parametri:** { confirmedrent\_id: int}

**Opis:** iz baze izbriše izposojo s podanim id-jem

# Rents

## GetAll

**Endpoint**: /rents

**Format**: “{user\_id}+{car\_id}+{fromdate}+{todate}+{completed}”

**Opis**: Izpiše vse izposoje iz baze.

## GetSpecific

**Endpoint**: / rents /getSpec

**Format**: “{user\_id}+{car\_id}+{fromdate}+{todate}+{completed}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določeni izposoji

## UpdateRent

**Endpoint**: / rents /updateRent

**Parametri**: {new\_id: int, new\_user\_id: int, new\_car\_id: int, new\_fromdate: String, new\_todate: String, new\_comfirmed: int}

**Opis**: Posodobi izposojo, ki ima id = new\_id

## InsertRent

**Endpoint**: / rents /insertRent

**Parametri**: {new\_user\_id: int, new\_car\_id: int, new\_fromdate: String, new\_todate: String }

**Opis**: V bazo vpiše novo izposojo

## DeleteRent

**Endpoint:** / rents /deleteRent

**Parametri:** {rent\_id: int}

**Opis:** iz baze izbriše izposojo s podanim id-jem

# Logs

## GetAll

**Endpoint**: /logs

**Format**: “{logdescription}+{logtype }”

**Opis**: Izpiše vse loge iz baze.

## GetSpecific

**Endpoint**: /logs/getSpec

**Format**: “{logdescription}+{logtype}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določenem logu

## UpdateLog

**Endpoint**: /logs/updateLog

**Parametri**: { new\_id: int, new\_logdescription: String, new\_logtype: int}

**Opis**: Posodobi log, ki ima id = new\_id

## InsertLog

**Endpoint**: /logs/insertLog

**Parametri**: { new\_logdescription: String, new\_logtype: int}

**Opis**: V bazo vpiše nov log

## DeleteLog

**Endpoint:** /logs/deleteLog

**Parametri:** {log\_id int}

**Opis:** iz baze izbriše log s podanim id-jem

# Garages

## GetAll

**Endpoint**: /garages

**Format**: “{name}+{city\_id }”

**Opis**: Izpiše vse garaže iz baze.

## GetSpecific

**Endpoint**: /garages/getSpec

**Format**: “{name}+{city\_id}”

**Parametri**: {id: int}

**Opis**: Izpiše podatke o določeni garaži

## UpdateGarage

**Endpoint**: /garages/updateGarage

**Parametri**: { new\_id: int, new\_name: String, new\_city\_id: int}

**Opis**: Posodobi garažo, ki ima id = new\_id

## InsertGarage

**Endpoint**: /garages/insertGarage

**Parametri**: { new\_name: String, new\_city\_id: int}

**Opis**: V bazo vpiše novo garažo

## DeleteGarage

**Endpoint:** /garages/deleteGarage

**Parametri:** {garage\_id int}

**Opis:** iz baze izbriše garažo s podanim id-jem