
PocketHome REST API

The PocketHome REST API is an API implemented to bridge the gap between the PostgreSQL database implemented in the PocketHome project and all the components that need to access it.

Houses

Information relative to the API interfaces that interact with the House entity in the database.

POST <http://localhost:8080/houses>

```
http://localhost:8080/houses
```

Store information about a new house in the database. ID's are generated automatically according to the entries already that already exist in the database.

HEADERS

Content-Type

application/json

BODY raw

```
{
  "address": "Avenida 29 de Março",
  "alias": "Casa Oliveira",
  "ownerId": 1
}
```

Example Request

```
http://localhost:8080/houses
```

```
curl --location --request POST 'http://localhost:8080/houses' \
--header 'Content-Type: application/json' \
--data-raw '{
  "address": "Avenida 29 de Março",
  "alias": "Casa Oliveira",
  "ownerId": 1
}'
```

GET http://localhost:8080/houses

http://localhost:8080/houses

Returns all the houses in the database

HEADERS

Content-Type

application/json

Example Request

http://localhost:8080/houses

```
curl --location --request GET 'http://localhost:8080/houses' \
--header 'Content-Type: application/json' \
--data-raw ''
```

PUT http://localhost:8080/houses/1

http://localhost:8080/houses/1

Edit information about a house in the database. You need to provide the ID of the house to be edited in the URL.

HEADERS

Content-Type

application/json

BODY raw

```
{
  "address": "Avenida 29 de Março",
  "alias": "Casa França",
  "ownerId": 1
}
```

Example Request

http://localhost:8080/houses/1

```
curl --location --request PUT 'http://localhost:8080/houses/1' \
--header 'Content-Type: application/json' \
--data-raw '{
  "address": "Avenida 29 de Março",
  "alias": "Casa França",
  "ownerId": 1
}'
```

Users

Information relative to the API interfaces that interact with the User entity in the database.

POST http://localhost:8080/users

http://localhost:8080/users

Adds a new user to the database. It is required to provide all the fields to store in the entry.

HEADERS

Content-Type

application/json

BODY raw

```
{
  "firstName": "Rodrigo",
  "lastName": "Oliveira",
  "email": "rfo08@ua.pt",
  "password": "123hello"
}
```

Example Request

```
http://localhost:8080/users
```

```
curl --location --request POST 'http://localhost:8080/users' \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "firstName": "Rodrigo",  
  "lastName": "Oliveira",  
  "email": "rfo08@ua.pt",  
  "password": "123hello"  
}'
```

GET http://localhost:8080/users

```
http://localhost:8080/users
```

Returns all the users in the database.

Example Request

```
http://localhost:8080/users
```

```
curl --location --request GET 'http://localhost:8080/users'
```

GET http://localhost:8080/users/1

```
http://localhost:8080/users/1
```

Returns information about a specific user in the database. You need to provide the ID of the user to be viewed in the URL.

Example Request

```
http://localhost:8080/users/1
```

```
curl --location --request GET 'http://localhost:8080/users/1'
```

PUT http://localhost:8080/users/1

```
http://localhost:8080/users/1
```

Edit information about a specific user in the database. You need to provide the ID of the user to be edited in the URL.

HEADERS

Content-Type

application/json

BODY raw

```
{
  "firstName": "Rodrigo",
  "lastName": "França",
  "email": "rfo08@ua.pt",
  "password": "123hello"
}
```

Example Request

```
http://localhost:8080/users/1
```

```
curl --location --request PUT 'http://localhost:8080/users/1' \
--header 'Content-Type: application/json' \
--data-raw '{
  "firstName": "Rodrigo",
  "lastName": "França",
  "email": "rfo08@ua.pt",
  "password": "123hello"
}'
```

Sensors

Information relative to the API interfaces that interact with the Sensor entity in the database.

POST http://localhost:8080/sensors

```
http://localhost:8080/sensors
```

Store information about a sensor in the database.

HEADERS

Content-Type

application/json

BODY raw

```
{
  "type": "DOOR_SENSOR",
  "houseId": 1
}
```

Example Request

```
http://localhost:8080/sensors
```

```
curl --location --request POST 'http://localhost:8080/sensors' \
--header 'Content-Type: application/json' \
--data-raw '{
  "type": "DOOR_SENSOR",
  "houseId": 1
}'
```

GET http://localhost:8080/sensors

```
http://localhost:8080/sensors
```

Returns all the existing sensors in the database.

Example Request

```
http://localhost:8080/sensors
```

```
curl --location --request GET 'http://localhost:8080/sensors'
```

GET http://localhost:8080/1/sensors

```
http://localhost:8080/1/sensors
```

Returns all the sensors that belong to a specific house in the database. You need to provide the houseId in the URL of the request.

Example Request

```
http://localhost:8080/1/sensors
```

```
curl --location --request GET 'http://localhost:8080/1/sensors'
```

GET http://localhost:8080/sensors/1

```
http://localhost:8080/sensors/1
```

Returns the information about a specific sensor in the database. You need to provide the sensorId in the URL of the request.

Example Request

```
http://localhost:8080/sensors/1
```

```
curl --location --request GET 'http://localhost:8080/sensors/1'
```

Sensor Logs

Information relative to the API interfaces that interact with the Sensor Log entity in the database.

POST http://localhost:8080/logs

```
http://localhost:8080/logs
```

Store a sensor log in the database. You need to specify all the necessary fields (the "img" field is facultative).

HEADERS

Content-Type

application/json

BODY raw

```
{
  "date": "2019-12-18 23:24:30.600",
  "sensorId": 8,
  "sensorType": "TEMP_SENSOR",
  "value": 30,
  "houseId": 1
}
```

Example Request

http://localhost:8080/logs

```
curl --location --request POST 'http://localhost:8080/logs' \
--header 'Content-Type: application/json' \
--data-raw '{
  "date": "2019-12-18 23:24:30.600",
  "sensorId": 8,
  "sensorType": "TEMP_SENSOR",
  "value": 30,
  "houseId": 1
}'
```

GET http://localhost:8080/logs

http://localhost:8080/logs

Returns all the sensor logs in the database.

Example Request

http://localhost:8080/logs


```
curl --location --request GET 'http://localhost:8080/logs'
```

GET http://localhost:8080/logs/1/8

```
http://localhost:8080/logs/1/8
```

Returns all the sensor logs relative to a certain house and sensorId. You need to provide the houseId and sensorId in the URL in the following format: "/logs/{house_id}/{sensor_id}"

Example Request

```
http://localhost:8080/logs/1/8
```

```
curl --location --request GET 'http://localhost:8080/logs/1/8'
```

GET http://localhost:8080/logs/1

```
http://localhost:8080/logs/1
```

Returns a specific sensor log in the database. You need to provide the Id of the log in the request URL.

Example Request

```
http://localhost:8080/logs/1
```

```
curl --location --request GET 'http://localhost:8080/logs/1'
```