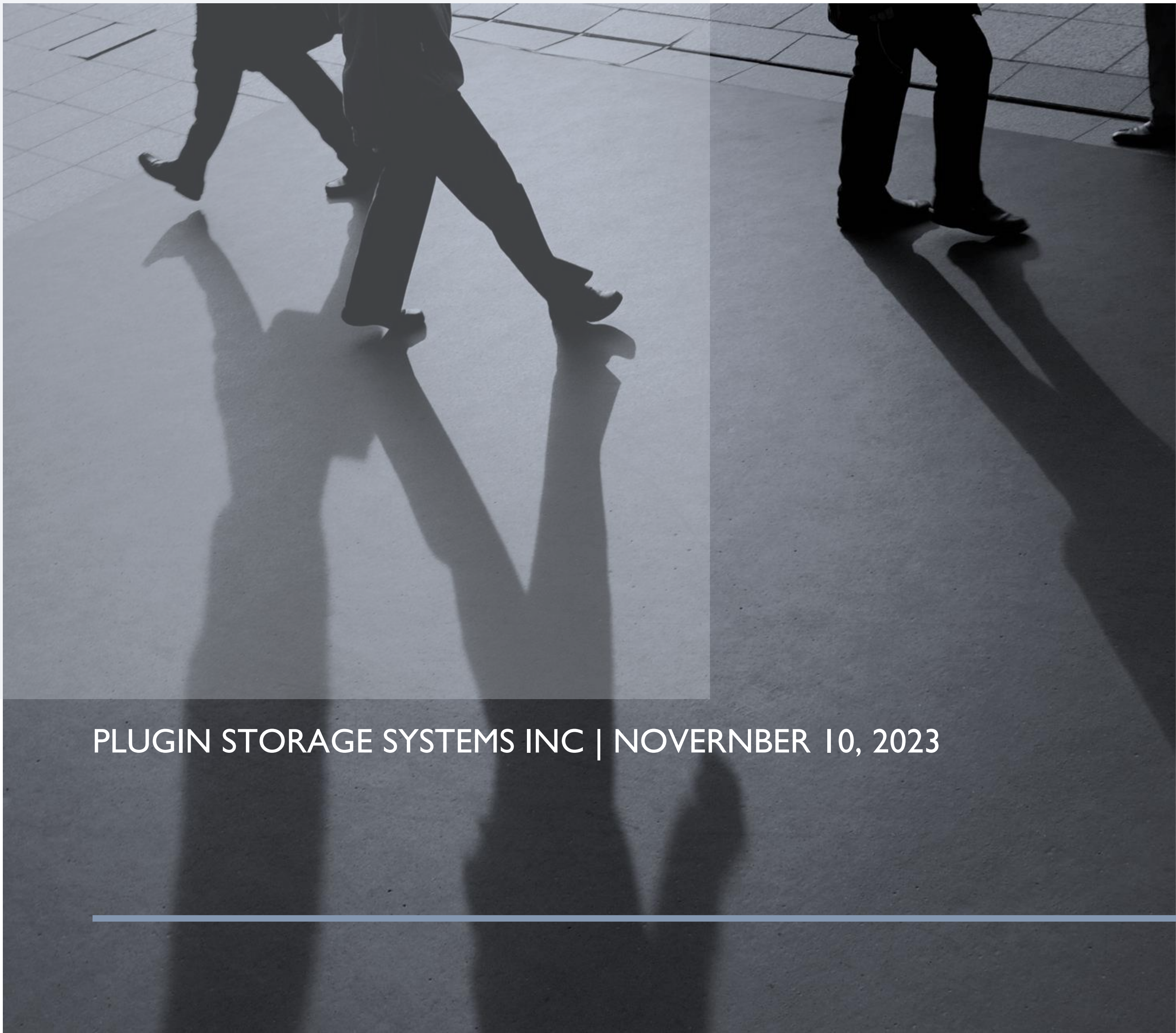
GUIDE OF

REST API

FOR PSSI CABINET



PLUGIN STORAGE SYSTEMS INC | NOVERNBER 10, 2023

TABLE OF CONTENTS

1 INTRODUCTION

1.1 API STYLE

Cabinet API allows the client to automate the cabinet management process. The APIs are exposed using JSON-RPC 2.0 protocol specified here:

http://www.jsonrpc.org/specification

The following is an example of the API:

```
{
  "id": 9,
  "UserName": "John Smith",
  "DrawerNum": 8
}
```

This API updates the use #9 with the “UserName” to “John Smith” and the “DrawerNum” to 8.

The software to verify the API in this document is Thunder Client on Visual Studio Code.

1.2 ACCOUNT AND AUTHENTICATION

A REST server is installed in the cabinet computer. The administrator of the server needs to create an account for himself or anyone who is using the APIs. The username and password for the account will be used when using the APIs.

The process of creating an account will be discussed in later chapters.

1.3 CLASSIFICATION OF APIS

The API can be classified as user management and cabinet configurations.

The types of API include GET, POST, PATCH. The action of GET is used to obtain the information from REST server, the action of POST is to add new information to the server, and the action of PATCH is to change the information.

2. USER MANAGEMENT

The cabinet administrator can use APIs to manage the users of the cabinet. The users can be added, modified or deleted.

2.1 LIST USER

This method is used to list all the users of the cabinet.

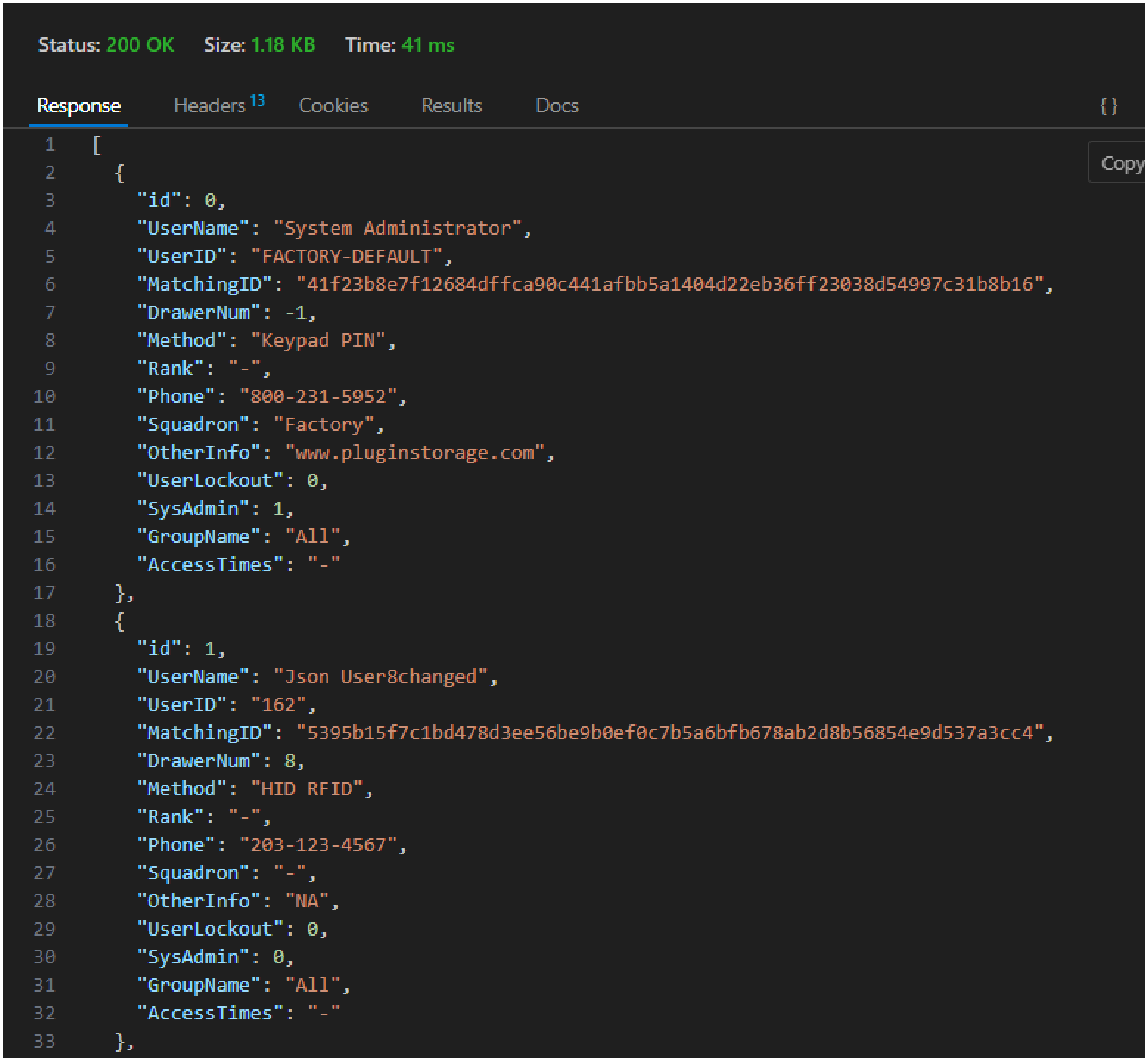
Option: GET

Address: http://<IP address>:<port number>/userlist/

JSON Content: None

Response: the detailed information of the users.

Example of response:



Status: 200 OK Size: 1.18 KB Time: 41 ms

```
[
  {
    "id": 0,
    "UserName": "System Administrator",
    "UserID": "FACTORY-DEFAULT",
    "MatchingID": "41f23b8e7f12684dffca90c441afbb5a1404d22eb36ff23038d54997c31b8b16",
    "DrawerNum": -1,
    "Method": "Keypad PIN",
    "Rank": "-",
    "Phone": "800-231-5952",
    "Squadron": "Factory",
    "OtherInfo": "www.pluginstorage.com",
    "UserLockout": 0,
    "SysAdmin": 1,
    "GroupName": "All",
    "AccessTimes": "-"
  },
  {
    "id": 1,
    "UserName": "Json User8changed",
    "UserID": "162",
    "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bfb678ab2d8b56854e9d537a3cc4",
    "DrawerNum": 8,
    "Method": "HID RFID",
    "Rank": "-",
    "Phone": "203-123-4567",
    "Squadron": "-",
    "OtherInfo": "NA",
    "UserLockout": 0,
    "SysAdmin": 0,
    "GroupName": "All",
    "AccessTimes": "-"
  },
```

2.2 ADD USERS

This method is used to add a user into the user list of the cabinet.

Option: POST

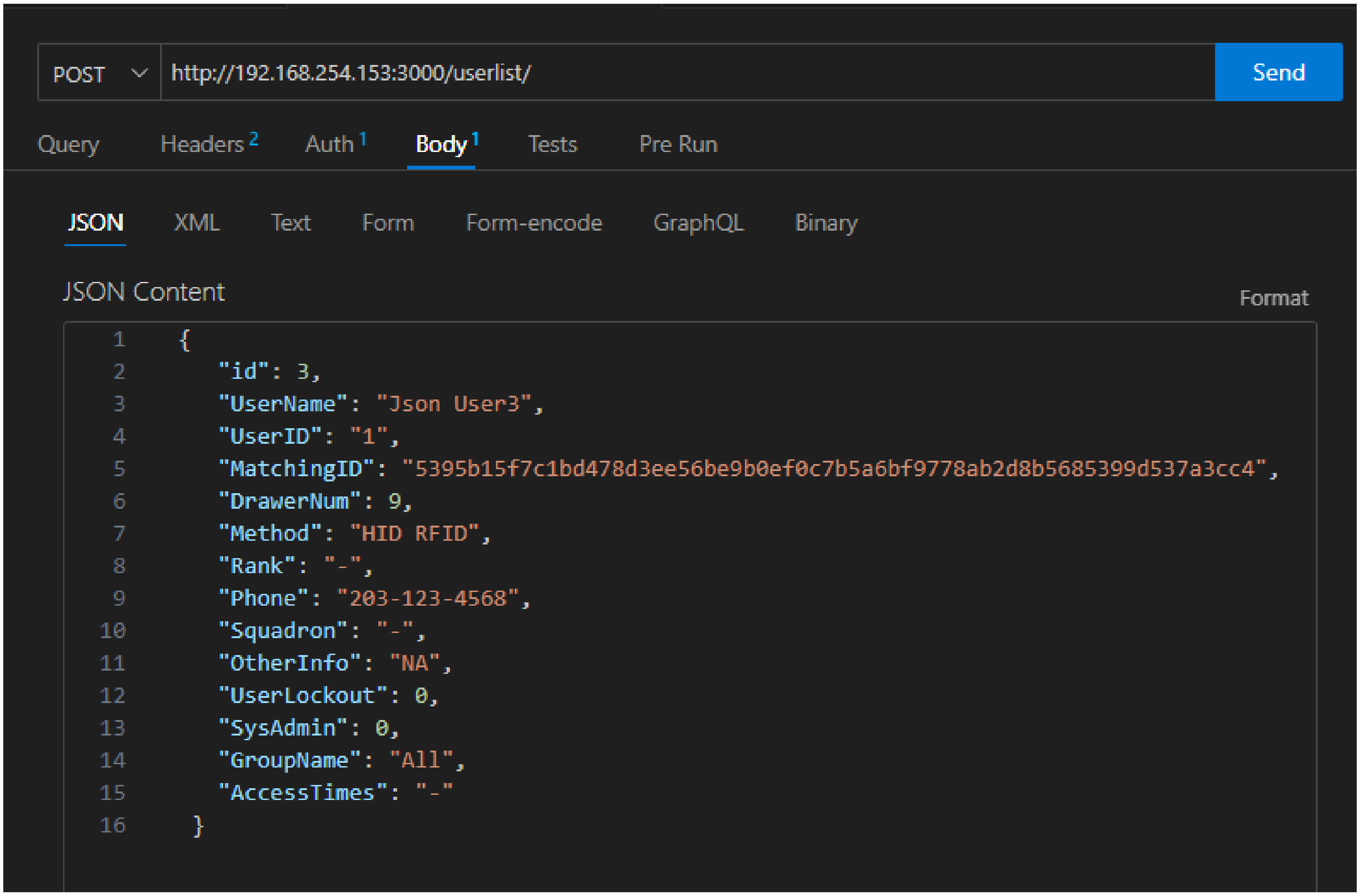
Address: http://<IP address>:<port number>/userlist/

JSON Content/Parameters:

Parameter	Type	Optional	Description
Id	String	No	the serial number of the record in JSON database
UserName	String	No	the name of the user
UserID	String	No	the user ID of the user in the cabinet database
MatchingID	String	No	the encrypted PIN number of the user
DrawerNum	Int	No	The assigned drawer for the user
Method	String	No	The method to access the cabinet, including HID, RFID, CacCard, PIN, and Barcode
Rank	String	Yes	The rank of the user
Phone	String	Yes	The phone number of the user
Squadron	String	Yes	The squadron of the user
OtherInfo	String	Yes	Other information of the user
UserLockout	Int	No	The locking status of the user. If it is "1", it means the user is locked out
SysAdmin	Int	No	The role of the user. If it is "1", it means the user is an administrator
GroupName	String	No	The group to which the user belongs. The default value is "All"
AccessTimes	String	Yes	The time that the user access the cabinet

Response: Display the record that has been added to the list. It is basically the same as the info in the request.

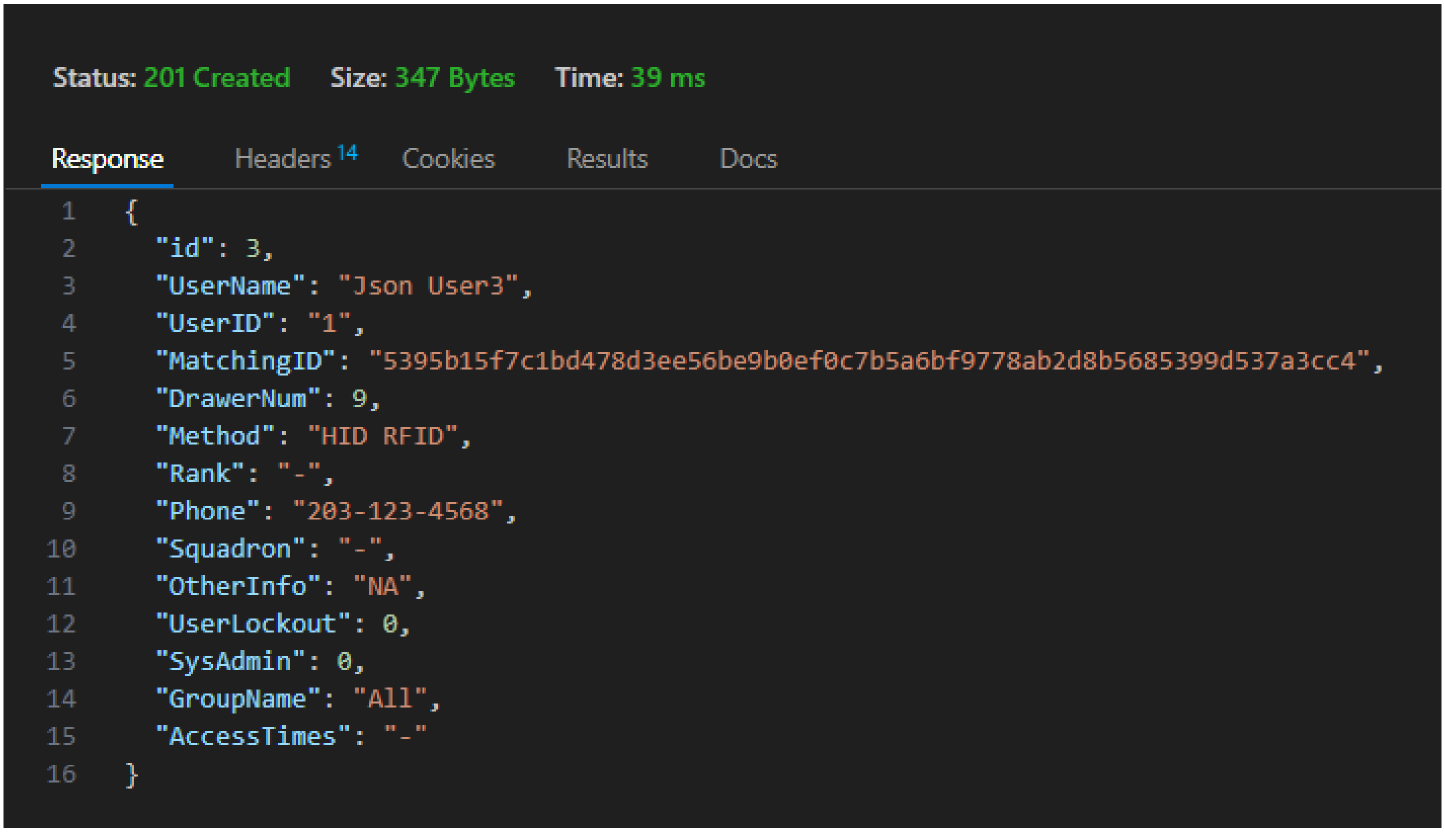
Example of the Request:



POST http://192.168.254.153:3000/userlist/

```
{
   "id": 3,
   "UserName": "Json User3",
   "UserID": "1",
   "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bf9778ab2d8b5685399d537a3cc4",
   "DrawerNum": 9,
   "Method": "HID RFID",
   "Rank": "-",
   "Phone": "203-123-4568",
   "Squadron": "-",
   "OtherInfo": "NA",
   "UserLockout": 0,
   "SysAdmin": 0,
   "GroupName": "All",
   "AccessTimes": "-"
 }
```

Example of the Response:



Status: 201 Created Size: 347 Bytes Time: 39 ms

```
{
  "id": 3,
  "UserName": "Json User3",
  "UserID": "1",
  "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bf9778ab2d8b5685399d537a3cc4",
  "DrawerNum": 9,
  "Method": "HID RFID",
  "Rank": "-",
  "Phone": "203-123-4568",
  "Squadron": "-",
  "OtherInfo": "NA",
  "UserLockout": 0,
  "SysAdmin": 0,
  "GroupName": "All",
  "AccessTimes": "-"
}
```

2.3 MODIFY USERS

This method is used to modify some information of the user. It uses the "id" to determine the user to be modified, and other information is changeable.

Option: PATCH

Address: http://<IP address>:<port number>/userlist/<id of the user>

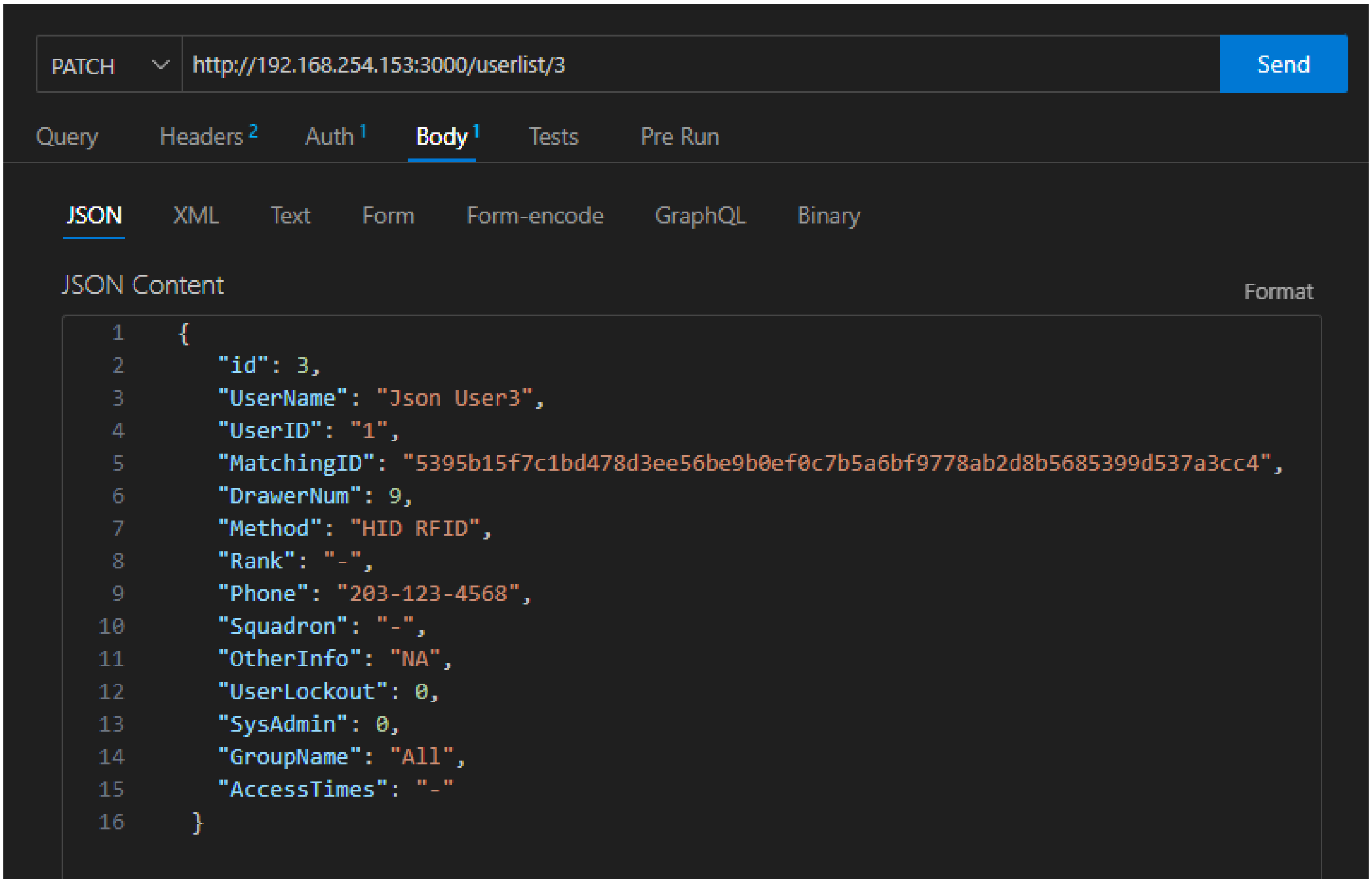
JSON Content/Parameters:

Parameter	Type	Optional	Description
Id	String	No	the serial number of the record in JSON database
UserName	String	Yes	the name of the user
UserID	String	Yes	the user ID of the user in the cabinet database
MatchingID	String	Yes	the encrypted PIN number of the user
DrawerNum	Int	Yes	The assigned drawer for the user
Method	String	Yes	The method to access the cabinet, including HID, RFID, CacCard, PIN, and Barcode
Rank	String	Yes	The rank of the user
Phone	String	Yes	The phone number of the user
Squadron	String	Yes	The squadron of the user
OtherInfo	String	Yes	Other information of the user
UserLockout	Int	Yes	The locking status of the user. If it is "1", it means the user is locked out
SysAdmin	Int	Yes	The role of the user. If it is "1", it means the user is an administrator
GroupName	String	Yes	The group to which the user belongs. The default value is "All"

AccessTimes	String	Yes	The time that the user access the cabinet

Response: Display the record that has been updated. It is basically the same as the info in the request.

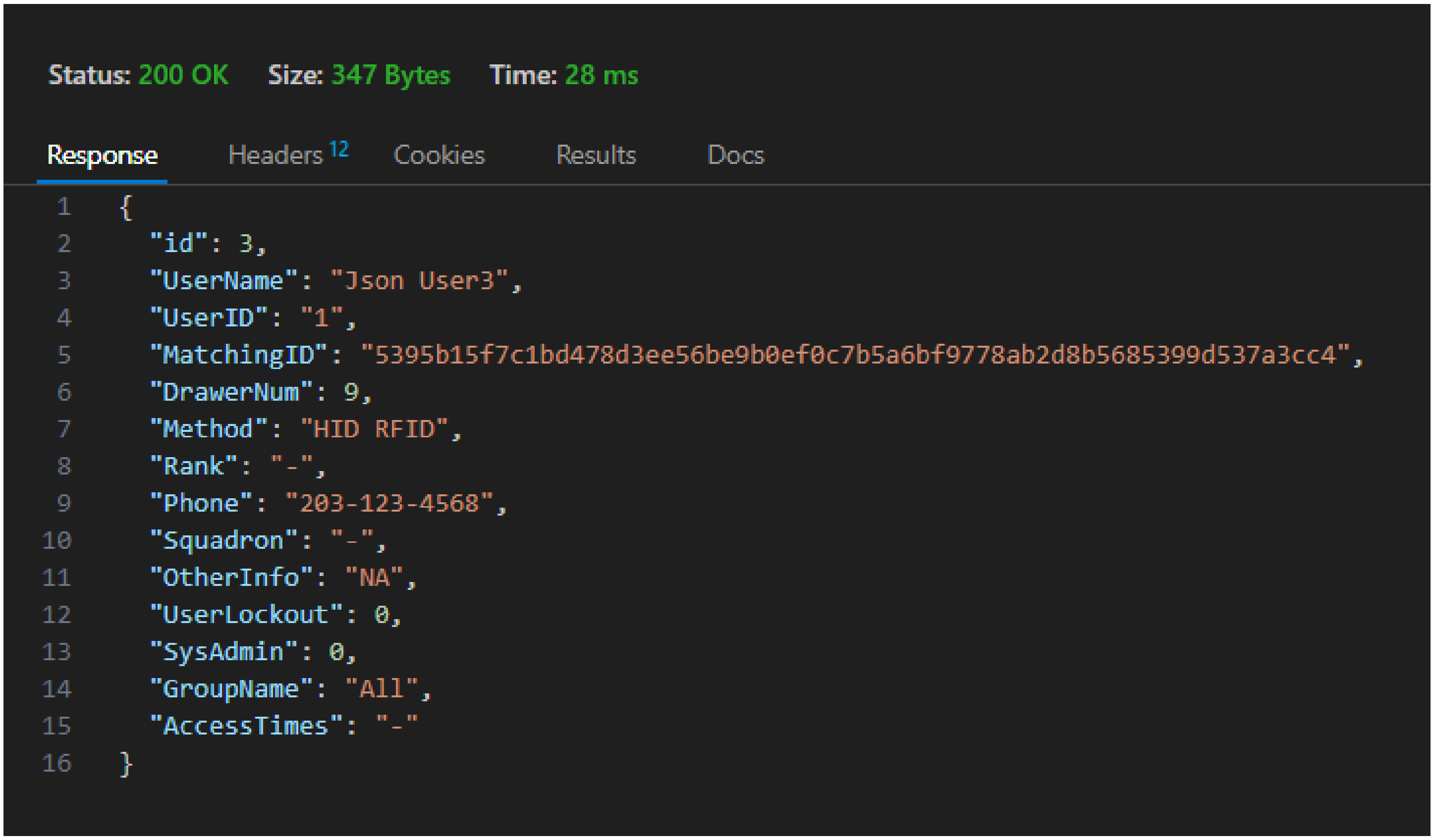
Example of the Request:



PATCH http://192.168.254.153:3000/userlist/3

```
{
   "id": 3,
   "UserName": "Json User3",
   "UserID": "1",
   "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bf9778ab2d8b5685399d537a3cc4",
   "DrawerNum": 9,
   "Method": "HID RFID",
   "Rank": "-",
   "Phone": "203-123-4568",
   "Squadron": "-",
   "OtherInfo": "NA",
   "UserLockout": 0,
   "SysAdmin": 0,
   "GroupName": "All",
   "AccessTimes": "-"
 }
```

Example of the Response:



Status: 200 OK Size: 347 Bytes Time: 28 ms

```
{
  "id": 3,
  "UserName": "Json User3",
  "UserID": "1",
  "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bf9778ab2d8b5685399d537a3cc4",
  "DrawerNum": 9,
  "Method": "HID RFID",
  "Rank": "-",
  "Phone": "203-123-4568",
  "Squadron": "-",
  "OtherInfo": "NA",
  "UserLockout": 0,
  "SysAdmin": 0,
  "GroupName": "All",
  "AccessTimes": "-"
}
```

2.4 REMOVE USERS

This method is used to delete a user from the user list. The "id" value of the user needs to be given to determine the user; other values of the user are optional.

Option: DELETE

Address: http://<IP address>:<port number>/userlist/<id of the user>

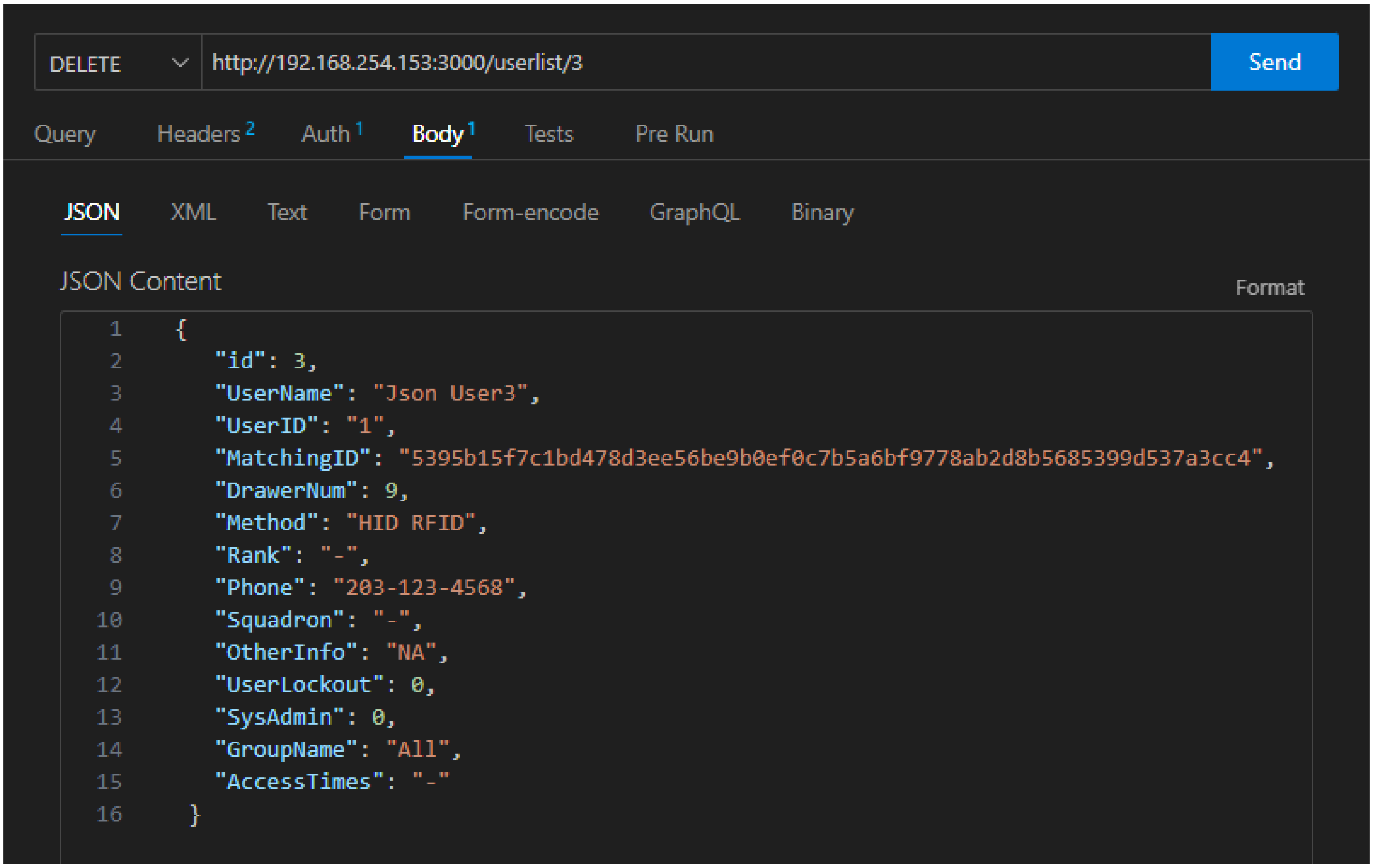
JSON Content/Parameters:

Parameter	Type	Optional	Description
Id	String	No	the serial number of the record in JSON database
UserName	String	Yes	the name of the user
UserID	String	Yes	the user ID of the user in the cabinet database
MatchingID	String	Yes	the encrypted PIN number of the user
DrawerNum	Int	Yes	The assigned drawer for the user
Method	String	Yes	The method to access the cabinet, including HID, RFID, CacCard, PIN, and Barcode
Rank	String	Yes	The rank of the user
Phone	String	Yes	The phone number of the user
Squadron	String	Yes	The squadron of the user
OtherInfo	String	Yes	Other information of the user
UserLockout	Int	Yes	The locking status of the user. If it is "1", it means the user is locked out
SysAdmin	Int	Yes	The role of the user. If it is "1", it means the user is an administrator

GroupName	String	Yes	The group to which the user belongs. The default value is “All”
AccessTimes	String	Yes	The time that the user access the cabinet

Response: Empty

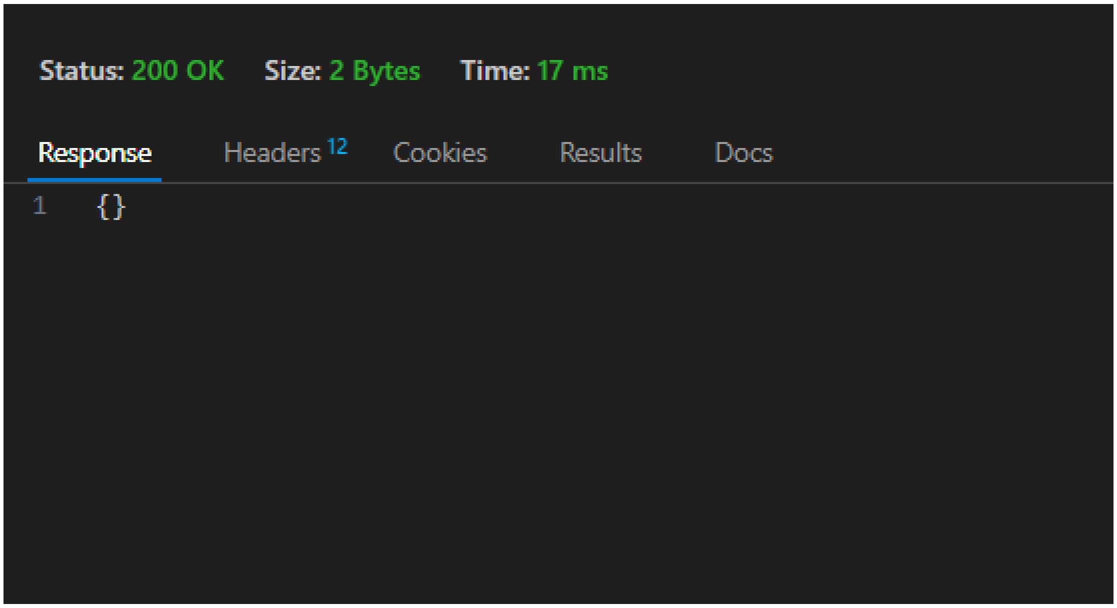
Example of the Request:



http://192.168.254.153:3000/userlist/3

```
{
   "id": 3,
   "UserName": "Json User3",
   "UserID": "1",
   "MatchingID": "5395b15f7c1bd478d3ee56be9b0ef0c7b5a6bf9778ab2d8b5685399d537a3cc4",
   "DrawerNum": 9,
   "Method": "HID RFID",
   "Rank": "-",
   "Phone": "203-123-4568",
   "Squadron": "-",
   "OtherInfo": "NA",
   "UserLockout": 0,
   "SysAdmin": 0,
   "GroupName": "All",
   "AccessTimes": "-"
 }
```

Example of the Response:



Status: 200 OK Size: 2 Bytes Time: 17 ms

```
{}
```

3. CABINET CONFIGURATION

The cabinet can be configured with REST API. Currently the client can change the access mode and the option to add users automatically.

3.1 THE SELECTION OF ADDING USER AUTOMATICALLY

The user can change the setting to turn on/off the function to add users automatically by RESTAPI.

Option: PATCH

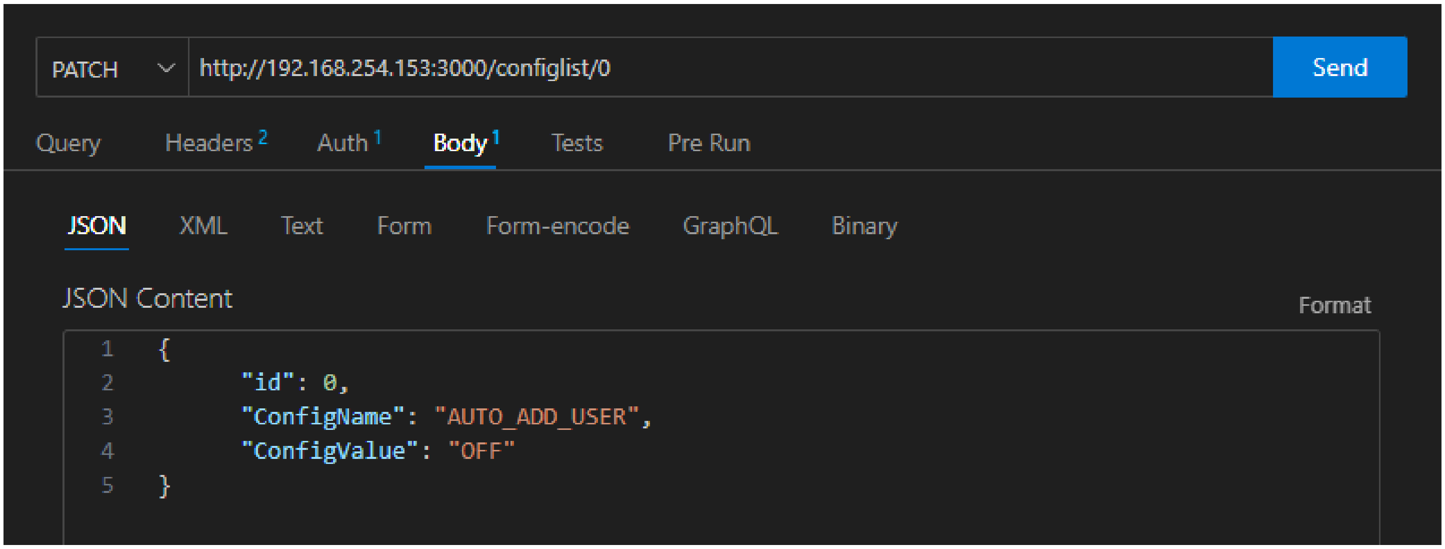
Address: http://<IP address>:<port number>/configlist/0

JSON Content/Parameters:

Parameter	Type	Optional	Description
Id	String	No	the serial number of the record in JSON database. It is set to "0".
ConfigName	String	Yes	The value is fixed: "AUTO_ADD_USER"
ConfigValue	String	No	The choice to turn on/off. It has two options: "ON" and "OFF"

Response: Display the record that has been updated. It is basically the same as the info in the request.

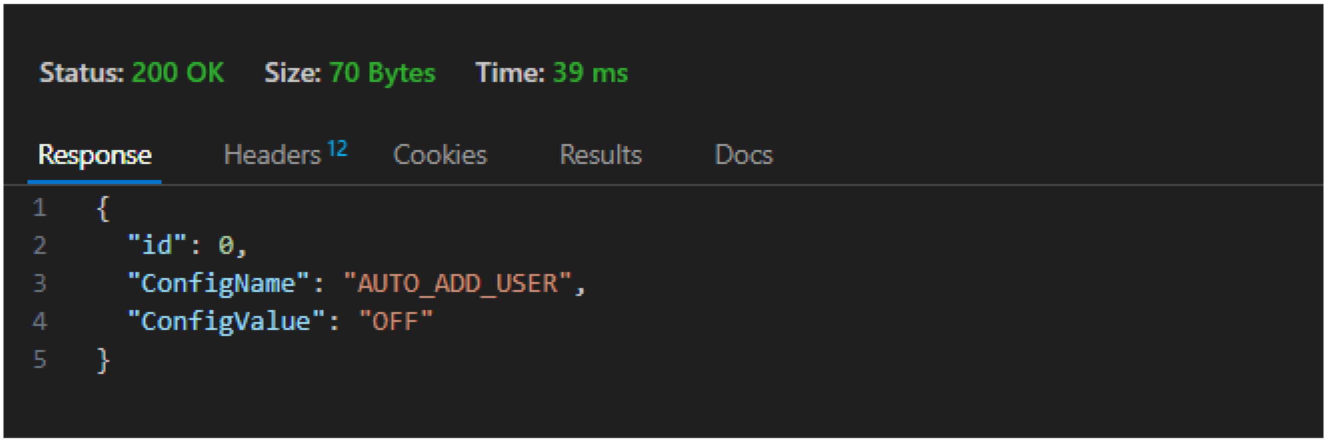
Example of the Request:



PATCH http://192.168.254.153:3000/configlist/0

```
{
       "id": 0,
       "ConfigName": "AUTO_ADD_USER",
       "ConfigValue": "OFF"
}
```

Example of the Response:



Status: 200 OK Size: 70 Bytes Time: 39 ms

```
{
  "id": 0,
  "ConfigName": "AUTO_ADD_USER",
  "ConfigValue": "OFF"
}
```

3.2 THE SELECTION FOR ACCESS MODE

The user can modify the access mode of the cabinet by using the REST API.

Option: PATCH

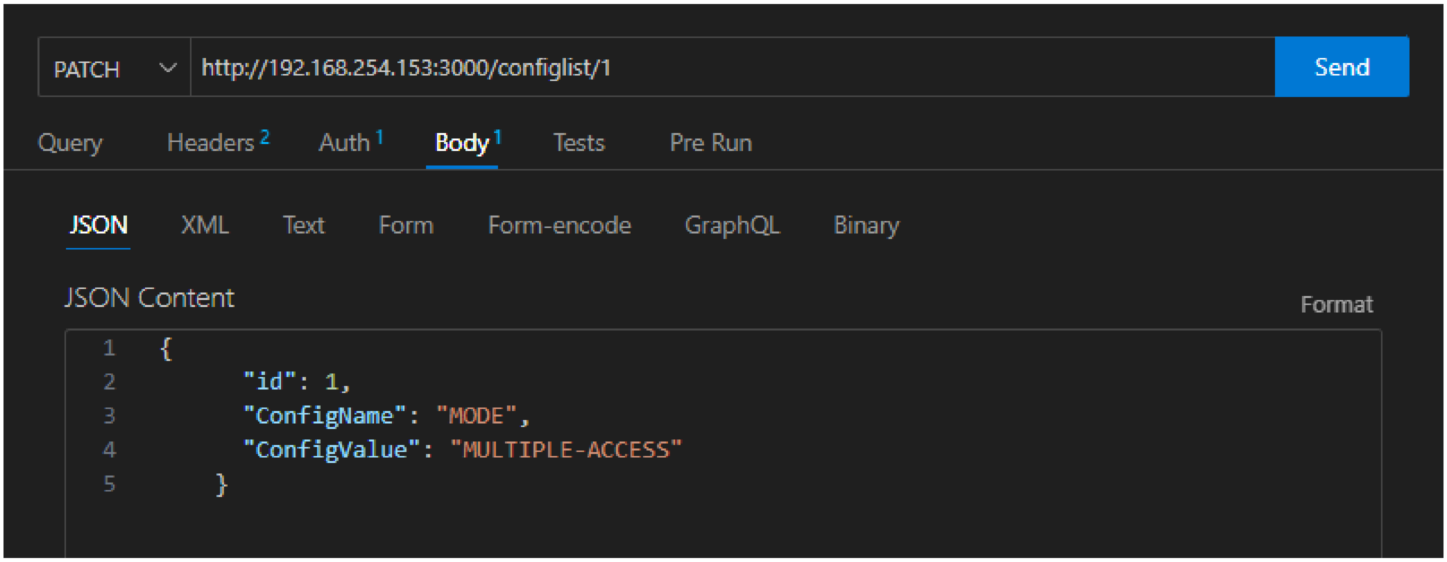
Address: http://<IP address>:<port number>/configlist/1

JSON Content/Parameters:

Parameter	Type	Optional	Description
Id	String	No	the serial number of the record in JSON database. It is set to "1"
ConfigName	String	Yes	The value is fixed: "MODE"
ConfigValue	String	No	The access mode will be one the following: FIRST-AVAILABLE, MULTIPLE-ACCESS, FIXED-ACCESS, CHECK-OUT-ONLY, CHECK-IN-ONLY, EQUIPMENT-SENSITIVE, ALL-ACCESS

Response: Display the record that has been updated. It is basically the same as the info in the request.

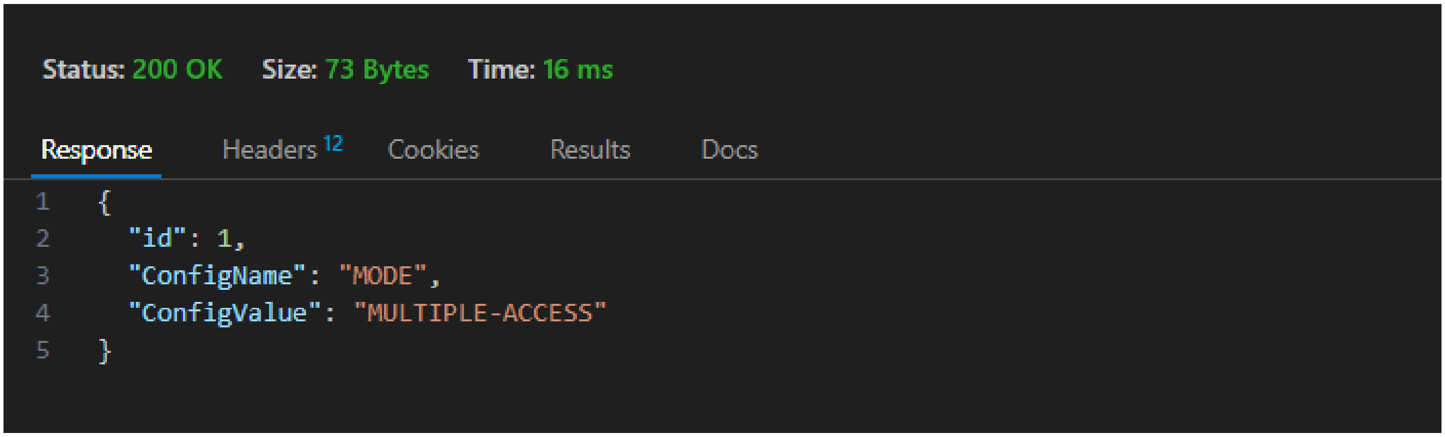
Example of the Request:



http://192.168.254.153:3000/configlist/1

```
{
        "id": 1,
        "ConfigName": "MODE",
        "ConfigValue": "MULTIPLE-ACCESS"
    }
```

Example of the Response:



Status: 200 OK   Size: 73 Bytes   Time: 16 ms

```
{
  "id": 1,
  "ConfigName": "MODE",
  "ConfigValue": "MULTIPLE-ACCESS"
}
```