



LOMIA SOFTWARE

LOmnia Keyless® is a QR Code engine / generator | Keypad codes to be installed locally on your own Server.
It allows you to generate QR Codes and Codes in full autonomy, even locally, to configure and use the following Keyless® products:

Okulo
Okulo II
Okulo PLUS
Okulo PICO
bEar-MHOM
Logger Wi-Fi

LOmnia Keyless® is licensed for use by signing a service contract valid for 12 calendar months.
The activation of the software and the license is annual and allows you to use and activate Keyless® devices (quantity depending of license type), listed above.
The generation of QR Codes | Codes is unlimited for the devices listed above.
Upon expiry of the license, if not renewed, it will no longer be possible to generate QR Code | Codes.

TECHNICAL REFERENCE

In order to function and be correctly configured, the LOmnia software must be installed on a Server with the following minimum characteristics:

Mac OS X with M1 chip
Mac OS X with Intel chip
Ubuntu Server 18 or higher
Windows 10 or higher
Linux
RAM 8 Gbyte
MySql database

Keyless® LOmnia is a multi-tread software



INTRODUCTION

Despite the attentive and careful realization, update and integration of the contents and integrations in this handbook, these contents and information may contain typing mistakes, inaccuracies or inconsistencies for whom Keyless® does not provide any guarantee - neither explicit nor implicit - not even by way of reliability or expectation. In particular, we do not guarantee completeness, exhaustiveness and compliance with the current organization, company structure or typology, characteristics, marketability and fitness for purpose of the products, services and activities of Keyless®, who is available to provide any relevant information.

Keyless® reserves the right to make adjustments and changes to the information contained in the handbook without prior notice, as well as to vary and to change the products, the services and the activities described herein, without it rising any kind of rights for a third-party.

TERMS AND CONDITIONS

Keyless® LOMnia API LICENSE AGREEMENT (Last revision early September 2022)

This is a legal agreement (the “Agreement”) between the you and, if applicable, your company, organization or other legal entity for whom you have authority to enter into this Agreement (“Developer”) and Keyless® (“Keyless®”), for use of certain of Keyless® Application Programming Interfaces (“LOmnia API”) that enable Developer to provide additional functionality for users of Keyless® products or to integrate access to certain Keyless® products into Developer’s product offerings. By clicking “I agree to the terms of service,” or otherwise accessing the LOMnia API, Developer indicates its acceptance of this Agreement.

1.1 Offering Integration; LOMnia API Documentation. During the term of this Agreement, Developer may use the LOMnia API to provide capabilities or integrations that leverage one or more the Keyless® products available at www.keyless.it (the “Keyless® Products”) into additional functionality, products, websites and/or services that are offered by Developer (the “Offerings”), subject to the terms and conditions of this Agreement. Keyless® may make available to Developer the Keyless® LOMnia API and any corresponding reference materials (including LOMnia API documentation, wrapper libraries, sample code and LOMnia API updates and changes) and source code, which may be amended or revised by Keyless® at any time (the “Keyless® LOMnia API Documentation”), for use for such purposes during the term of this Agreement.

1.2. Developer Information; Consent to Contact. Developer shall provide Keyless® with Developer’s contact information and hereby consents to Keyless® sharing such information with any User (defined below) or prospective User of the Offering. Developer shall update the contact information, as needed, such that Keyless® always has current contact information for the Developer and the Offering. By entering into this Agreement, Developer consents to receiving phone calls, emails, texts or any other type of messages from Keyless® to inform it of changes or additions to the Keyless® Products, this Agreement, the LOMnia API or the LOMnia API Documentation and any other matter related to the foregoing and for general marketing purposes (Developer may unsubscribe from marketing messages at any time, but not transactional messages). Keyless® may, but is not obligated to, monitor or record any telephone conversations and chat texts for quality control purposes, for purposes of training employees and for Keyless®’s own protection.

1.3 User Terms and Conditions. Developer acknowledges and agrees that users of the Offerings (“Users”) must be Keyless® account holders for a Keyless® Product in order to access and use such Keyless® Product through the Offerings. Developer further acknowledges and agrees that all Users’ access and use of the Keyless® Products is subject, in all respects, to the User Agreements. Developer hereby agrees to either: (i) require all Users to agree to the User Agreements prior to permitting such users to access the Keyless® Products through the Offerings; or (ii) only provide access to Keyless® Products in a manner that requires Users to register with Keyless® directly and accept the User Agreements. Developer further agrees that, to the extent Developer uses the Keyless® Products on its own behalf or on behalf of its customers, Developer’s use shall be subject to the User Agreements in all respects. Developer agrees that it may use the LOMnia API to add or update only customers that have been obtained by the User using permission-based standards that meet standards described in the User Agreements. The “User Agreements” are defined as Keyless® Web Site and Products Terms and Conditions of Use, Privacy Statement, Anti-Spam Policy, and any other acceptable use policy, content restrictions, user agreements, and other terms and conditions governing use of the Keyless® Products, generally available through the Keyless® websites, as each of the foregoing may be amended by Keyless® from time to time in its sole discretion.

1.4 Privacy Policy; Customer Accounts. Developer represents, warrants and covenants that, in its operation of the Offerings, it will maintain and comply with a privacy policy that complies with applicable law and that accurately discloses how Developer collects, uses, stores, and discloses data provided by Users and third parties. Developer will post its privacy policy in the Offerings. Further, if Developer has any login, customer or other account information relating to a User’s account with Keyless®, Developer shall only use such information for the purposes expressly authorized by the applicable User.

1.5 Responsibility for Offerings. Developer is solely responsible for the Offerings and Keyless® shall have no liability or obligations with respect to the same (including support obligations). Developer represents, warrants and covenants that Developer has and will at all times maintain the right to provide all Offerings provided by Developer hereunder and that the Offerings (and any other materials provided to Keyless® or Users) do not infringe the intellectual property or other rights of any third parties or contain viruses, worms, malware or any other harmful scripts or code. Developer agrees to provide support for its Offerings. Developer shall ensure that all Offerings that access the Keyless® Products comply with all applicable laws and regulations. Developer agrees to place the following notice prominently in the Offering: “This product uses the Keyless® LOMnia API but is not endorsed or certified by Keyless®.”

1.6 Fees. The LOMnia API are currently provided within LOMnia Software Subscription.

2. Licenses, Responsibilites and Restrictions.

2.1 License. Keyless® hereby grants to Developer a revocable, non-exclusive, non-transferable, limited license to use and integrate the LOMnia API into the Offering for the sole purpose of developing the integration to the Offering and allowing access to the Keyless® Products via the Offering. Once integrated into the Offering, access to the LOMnia API may then be distributed to the Users as an integrated part of the Offering.

2.2 Responsibilities; Restrictions. Developer shall implement the LOMnia API in accordance with the Keyless® LOMnia API Documentation. Developer may not access the LOMnia API if Developer is a competitor of Keyless®, as determined by Keyless® in its reasonable discretion, or to replicate or attempt to replicate the essential user experience of the Keyless® Products. Except as expressly provided herein, Developer has no other right to install, integrate, use, reproduce, sublicense or distribute LOMnia API.

Developer shall not: (i) modify, reverse engineer, decompile, or otherwise alter or attempt to gain access to the LOMnia API or the Keyless® Products in a manner not in accordance with this Agreement, (ii) use or enable its customers to use the LOMnia API for the purposes of testing or comparison of Keyless® Products or for any purpose competitive with Keyless® Products, or (iii) perform bulk operations with LOMnia API that are designed for single contact operations or perform single contact operations with LOMnia API that are designed for performing bulk operations. Developer agrees to protect the security and confidentiality of any credentials and LOMnia API keys disclosed by Keyless® hereunder.

2.3 Ownership; No Other Licenses. The LOMnia API contained in the Offering shall remain the sole and exclusive intellectual property of Keyless® and Developer shall reasonably assist Keyless® in protecting such ownership. No other licenses or rights in any of Keyless®'s intellectual property rights are granted hereunder. For example and without limitation, no rights are granted to use Keyless®'s logos or trademarks; provided, however that Developer may refer to the names of the Keyless® Products solely for the purpose of describing the Offering.

2.4 Right to Developer's Ideas, Logo and Name. Developer hereby grants to Keyless® a non-exclusive, royalty-free, irrevocable, perpetual license (i) to use any ideas that Keyless® learns from observing Developer's Offerings or other use of the LOMnia API or any feedback provided by Developer and (ii) to use Developer's name and logo for the purpose of disclosing that Developer is providing Offerings using the LOMnia API and for promotion of the availability of the LOMnia API.

3. Non-Exclusivity. Developer acknowledges that Developer's right to use and demonstrate the Keyless® Products hereunder is non-exclusive, and that Keyless® reserves the right to sell and distribute any of its services to any customers in the world, and to appoint any third party to do so, without giving Developer notice thereof and without incurring any liability to Developer therefore. Keyless® reserves the right to develop and extend its products and capabilities without regard to whether those products compete with or invalidate any Developer Offering. Unless otherwise mutually agreed by the parties, Keyless® may contact directly any User for the purpose of marketing and selling the Keyless® Products. Unless otherwise mutually agreed by the Parties, in the event that such User elects to purchase the Keyless® Products, Keyless® shall have no obligation to Developer with respect to such transaction.

4. Indemnification; Limitation of Liability; Disclaimer.

4.1 Indemnification. Developer shall defend, indemnify and hold Keyless® and its underlying service providers, business partners, third-party suppliers and providers, members of its network, account providers, licensors, officers, directors, employees, distributors and agents harmless from and against any lawsuit, claim, damage, liability, or expense (including reasonable attorneys' fees) incurred by Keyless® as a result of any third-party claim against Keyless® resulting from or relating to the Offering, Developer's use of the LOMnia API, Keyless®'s use of the Developer's marks, the content on Developer's website, Developer's unauthorized marketing, promotion, use or distribution of the Keyless® Products, Developer's failure to abide by the applicable terms of any User Agreement, Developer's breach of this Agreement, or the infringement or misappropriation of any patent, copyright, trademark, or other intellectual property right of any third party.

4.2 Limitation of Liability. EXCEPT WITH RESPECT TO DEATH OR PERSONAL INJURY DUE TO THE NEGLIGENCE OF Keyless®, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL Keyless® OR ANY OF ITS UNDERLYING SERVICE PROVIDERS, BUSINESS PARTNERS, THIRD PARTY SUPPLIERS AND PROVIDERS AND MEMBERS OF ITS NETWORK, ACCOUNT PROVIDERS, LICENSORS, OFFICERS, DIRECTORS, EMPLOYEES, DISTRIBUTORS OR AGENTS (COLLECTIVELY REFERRED TO FOR PURPOSES OF THIS SECTION AS "Keyless®") BE LIABLE TO DEVELOPER OR ANY OTHER PERSON FOR ANY MONEY DAMAGES, WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, RELIANCE OR CONSEQUENTIAL DAMAGES, EVEN IF Keyless® SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY, AND REGARDLESS OF THE FORM OF ACTION (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE), THE MAXIMUM AGGREGATE LIABILITY OF Keyless® TO DEVELOPER ARISING IN CONNECTION WITH THIS AGREEMENT SHALL BE LIMITED TO Euro 100,00. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY.

Developer agrees that Keyless® has made the LOMnia API available and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that they reflect an allocation of risk between the parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that they form an essential basis of the bargain between the parties.

4.3 Disclaimer. DEVELOPER EXPRESSLY AGREES THAT THE LOMnia API AND THE Keyless® PRODUCTS ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. USE OF THE LOMnia API OR THE Keyless® PRODUCTS AND ANY RELIANCE BY DEVELOPER UPON THE LOMnia API OR THE Keyless® PRODUCTS, INCLUDING ANY ACTION TAKEN BY DEVELOPER BECAUSE OF SUCH USE OR RELIANCE, IS AT DEVELOPER'S SOLE RISK. Keyless® DOES NOT WARRANT THAT THE USE OF THE LOMnia API OR THE Keyless® PRODUCTS WILL BE UNINTERRUPTED OR ERROR FREE, NOR DOES Keyless® MAKE ANY WARRANTY AS TO THE RESULTS THAT MAY BE OBTAINED FROM USE OF THE SAME. Keyless® DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. NO STATEMENT OR INFORMATION, WHETHER ORAL OR WRITTEN, OBTAINED FROM Keyless® IN ANY MEANS OR FASHION SHALL CREATE ANY WARRANTY NOT EXPRESSLY AND EXPLICITLY SET FORTH IN THIS AGREEMENT. Keyless® MAY MODIFY OR TERMINATE OR RESTRICT ACCESS TO THE LOMnia API AT ANY TIME WITHOUT NOTICE.

Keyless® SHALL HAVE NO LIABILITY OF ANY NATURE WHATSOEVER FOR DEVELOPER'S COMPLIANCE WITH OR BREACH OF ANY LICENSE OR TERMS AND CONDITIONS OF ANY THIRD PARTIES OR THIRD PARTY SERVICES.

NO CLAIM MAY BE ASSERTED BY DEVELOPER AGAINST Keyless® MORE THAN 12 MONTHS AFTER THE DATE OF THE CAUSE OF ACTION UNDERLYING SUCH CLAIM. DEVELOPER'S SOLE AND EXCLUSIVE REMEDY FOR ANY FAILURE OR NONPERFORMANCE OF THE LOMnia API OR THE PRODUCTS SHALL BE FOR Keyless® TO USE COMMERCIALY REASONABLE EFFORTS TO ADJUST OR REPAIR THE LOMnia API OR THE PRODUCTS.

5. Term; Termination. Developer may terminate this Agreement at any time by ceasing to provide Offerings and sending a confirmatory e-mail to the Web Services Coordinator (assistenza@keyless.it). Keyless® may terminate this Agreement and/or disable Developer's ability to provide Offerings via the LOMnia API, in each case at any time with or without cause, and with or without notice. Keyless® shall have no liability to Developer or any third party because of such termination or action. This Agreement terminates automatically if Developer breaches any term of this Agreement. The following provisions shall survive expiration or termination of this Agreement: Sections 1 (Enabling Access; Developer Obligations), 3 (Non-Exclusivity), 4 (Indemnification; Limitation of Liability; Disclaimer), 5 (Term; Termination) and 7 (Miscellaneous).

6. Restricted Persons; Export of Products or Technical Data. Developer hereby warrants that Developer is not a Restricted Person. For purposes of this Agreement, Developer is a Restricted Person if Developer or any officer, director, or controlling shareholder of Developer is (i) a national of or an entity existing under the laws of any country with which Italian persons are prohibited from engaging in transactions, as may be determined from time to time by Italy Treasury Department; (ii) designated as a Specially Designated National or institution of primary money laundering concern by the Italy State Treasury Department; (iii) listed on the Denied Persons List or Entity List by Italy Commerce Department; (iv) engaged in nuclear, missile, chemical or biological weapons activities to which Italian persons may not contribute without a Italian Government license; or (v) owned, controlled, or acting on behalf of a Restricted Person.

If Developer becomes a Restricted Person during the term of this Agreement, Developer shall notify Keyless® (info@keyless.it) within twenty-four (24) hours, and Keyless® shall have the right to terminate any further obligations to Developer, effective immediately and with no further liability to Developer, but without prejudice to Developer's outstanding obligations to Keyless®. Developer agrees that Developer shall not utilize the Products to conduct or facilitate any transaction with any Restricted Person, except as may be expressly authorized in advance in writing by the Italian Government. Developer may not remove or export from the Italy or allow the export or re-export of the Products, or any direct product thereof, including technical data, in violation of any restrictions, laws, or regulations of Italy or any other applicable country.

7. Miscellaneous

7.1 Full Force and Effect. If any provision of this Agreement is found to be unenforceable or invalid, that provision will be limited or eliminated to the minimum extent necessary so that this Agreement will otherwise remain in full force and effect and enforceable.

7.2 Entire Agreement. Keyless® and Developer agree that this Agreement is the complete and exclusive statement of the mutual understanding of the parties and supersedes and cancels all previous written and oral agreements, communications, and other understandings relating to the subject matter of this Agreement, and that this Agreement may be amended from time to time by Keyless® with or without advance notice to Developer. No delay or omission by either party in exercising any right or remedy under this Agreement or existing at law or equity shall be considered a waiver of such right or remedy. In the event of any inconsistency between this Agreement and the User Agreements, this Agreement controls.

7.3 Assignment. Developer may not assign any of its rights hereunder. Keyless® may assign all rights to any other individual or entity in its sole discretion.

7.4 Further Assurances. Developer agrees to execute any and all documents and take any other actions reasonably required to effectuate the purposes of this Agreement.

7.5 Third Party Beneficiaries. Keyless® underlying service providers, business partners, third-party suppliers and providers, members of its network, account providers, licensors, officers, directors, employees, distributors and agents are expressly made third party beneficiaries of this Agreement. Except as set forth in the immediately preceding sentence, nothing express or implied in this Agreement is intended to confer, nor shall anything herein confer, upon any person other than the parties and the respective permitted successors or assigns of the parties, any rights, remedies, obligations or liabilities whatsoever.

7.6 Titles. The titles of the paragraphs of this Agreement are for convenience only and have no legal or contractual effect.

7.7 No Agency. Except as expressly set forth herein, no agency, partnership, joint venture, or employment is created as a result of this Agreement, and Developer does not have any authority of any kind to bind Keyless® in any respect whatsoever.

7.8 Attorney Fees. In any action or proceeding to enforce rights under this Agreement, the prevailing party will be entitled to recover its costs and attorneys' fees.

7.9 Authority. Developer represents that Developer has the full power, capacity and authority to accept this Agreement. If Developer is accepting on behalf of its employer or another entity, Developer represents that it has full legal authority to bind its employer or such entity to this Agreement.

7.10 Governing Law and Legal Actions. This Agreement shall be governed by the laws of Italy, and all claims relating to or arising out of this Agreement, or the breach thereof, whether sounding in contract, tort or otherwise, shall likewise be governed by the laws of the State of Italy, in each case, without regard to its choice or law or conflict of laws provisions. All legal actions in connection with this Agreement shall be brought in the state or federal courts located in Treviso, Italy.

7.11 Equitable Relief. Developer agrees that any violation or threatened violation of this Agreement may cause irreparable injury to Keyless®, entitling Keyless® to obtain injunctive or other equitable relief in addition to all legal remedies.

LOmnia

API

CHAPTER 0

HOW TO & SETUP

HOW TO GUIDE

LOmnia Keyless® must be installed on a Server reachable by your production software that interrogates and manages it.

If installed on a remote Server, port 1211 on the Router must be opened and the Router itself must be reachable at its IP address.

If necessary, consult a network and systems expert.

In the resident Server, MySql must also be installed and properly configured.

It is recommended that you set up an automatic database backup.

Suggestion:

a classic backup should follow this logic

hourly

every 12 hours

every 24 hours

every week

every month

naming each single backup with date, time and incremental number

Attention: LOmnia sets its own date and time based on the Server on which it is installed.

It is advisable to perform a cyclic routine that queries the online NTP Servers and updates the date and time of the Server on which it is installed.

It is strongly recommended to run LOmnia using third-party software such as: PM2

In the event of a crash or software that freezes, the third-party software (PM2) tries to restart the system indefinitely.

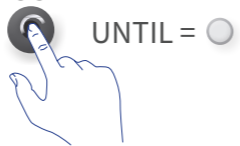



Also you can create "Warning" alerts in this scenario.

LOmnia Keyless® has been extensively tested under extreme conditions.

It is able to support 150K requests to generate QR Code - Codes in 10 seconds (capacity also dependent on the hardware on which it is installed).

The Token for the API is located inside the Json file in the Lomnia folder

FACTORY RESET - OKULO II & OKULO PLUS II

Action	Display
<p>Okulo II central unit (button located at the bottom right): With the system OFF, press and hold the reset button, power on while ALWAYS pressed the button (until the white LED flashes)</p>	<p>HOLD PUSH UNTIL = LED BLINK</p> 
<p>WiFi Logger (button located in the top center): With the system OFF, press and hold the reset button, power on while ALWAYS pressed the button (until ALL the WiFi signal LEDs come on)</p>	<p>HOLD PUSH UNTIL = ALL LED ON</p> 
Release the button	
<p>Wait for the final melody. The central unit is reset to the factory settings</p>	<p>LED OFF + </p>



Please note

Factory reset restores the ECU to its original settings

It does not reset instead:

- Date and time
- Black List

SET THE CONTROL UNIT BY FOLLOWING THE INDICATED PROCEDURE

Action	Display
<p>Press the RESET buttons (see procedure on previous page) Scan the generated QR Codes</p>	<p>-></p>
<p>Set the following parameters from the Account:</p> <ol style="list-style-type: none"> 1. CREATE A GROUP 2. CREATE OR MODIFY A CENTRAL UNIT 3. SET DATE & TIME 4. RESET KEYPAD MEMORY 5. RESET BLACKLIST MEMORY 6. SET LOGGER WIFI 7. CREATE A PERSONAL POST PATH FOR CENTRAL UNIT 	
<p>Generate QR Codes with these settings</p>	
<p>Wait for the final melody for each scanned QR Code The controller is set up and ready to operate</p>	

Please note

The control unit, once set, for safety reasons, does not accept further settings.

To change the Control unit Setup and Wi-Fi Setup data, it must be reset to bring it back to factory conditions

FACTORY RESET - OKULO PICO

Action	Display
<p>With the system OFF, join with a cable / paper clip the two RESET holes, apply tension by ALWAYS holding the two holes together</p> <p>(until ALL LEDs and MELODY turn on)</p>	
<p>Release</p>	
<p>Wait for the final melody (the LEDs go off). The controller is reset to the factory settings</p>	<p>LED OFF +</p>

Please note

Factory reset restores the central unit to its original settings

It does not reset instead:

- Date and time
- Black List
- Association of the control unit to your PLAFOND / ACCOUNT

SET THE CONTROL UNIT BY FOLLOWING THE INDICATED PROCEDURE

Action	Display
<p>Join the RESET holes (see procedure on previous page) Scan the generated QR Codes</p>	<p>-></p>
<p>Set the following parameters from the Account:</p> <ol style="list-style-type: none"> 1. CREATE A GROUP 2. CREATE OR MODIFY A CENTRAL UNIT 3. SET DATE & TIME 4. RESET BLACKLIST MEMORY 5. SET LOGGER WIFI 6. CREATE A PERSONAL POST PATH FOR CENTRAL UNIT <p>Generate QR Codes with these settings</p>	
<p>Wait for the final melody for each scanned QR Code The central unit is set up and ready to operate</p>	

Please note

The central unit, once set, for safety reasons, does not accept further settings.

To change the central unit Setup and Wi-Fi Setup data, it must be reset to bring it back to factory conditions

LOmnia API

CHAPTER 1 OKULO & MHOM



OKULO



This chapter contain all instruction for Okulo, Okulo II, Okulo PICO, MHOM version.

INTRODUCTION

The API are reachable via REST API which return a JSON as a response.
All calls must be authenticated using your access TOKEN.



CREATE A GROUP



NOTICE



Each group can contain a maximum of 96 control units with the same or different passwords
Infinite Groups can be created

FUNCTION: CREATE A GROUP

Method: **POST**

URL: `http://{ip}:1211/groups/create`

Response: **JSON**

Keyless[®]

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
name	Name of Group	STRING (12 characters Fixed including blanks)	Mandatory
note	Note of Group	STRING LONG TEXT	Optional

DELETE A GROUP



NOTICE



Pay attention to the group ID.
query the database to verify the ID of the group candidate for deletion

FUNCTION: CREATE A GROUP

Method: **POST**
URL: `http://{ip}:1211/groups/delete`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
id	ID of Group	INTEGER	Mandatory





Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

FUNCTION: LIST OF ALL GROUPS

Method: **POST**
URL: http://{ip}:1211/groups
Response: **JSON**

Keyless[®]

CREATE OR MODIFY A CENTRAL UNIT



NOTICE



If the central unit already exists, it overwrites / updates its values, otherwise it creates it
Warning: the ID of the control units can be freely set even if not sequentially from the value 1 to the value 96

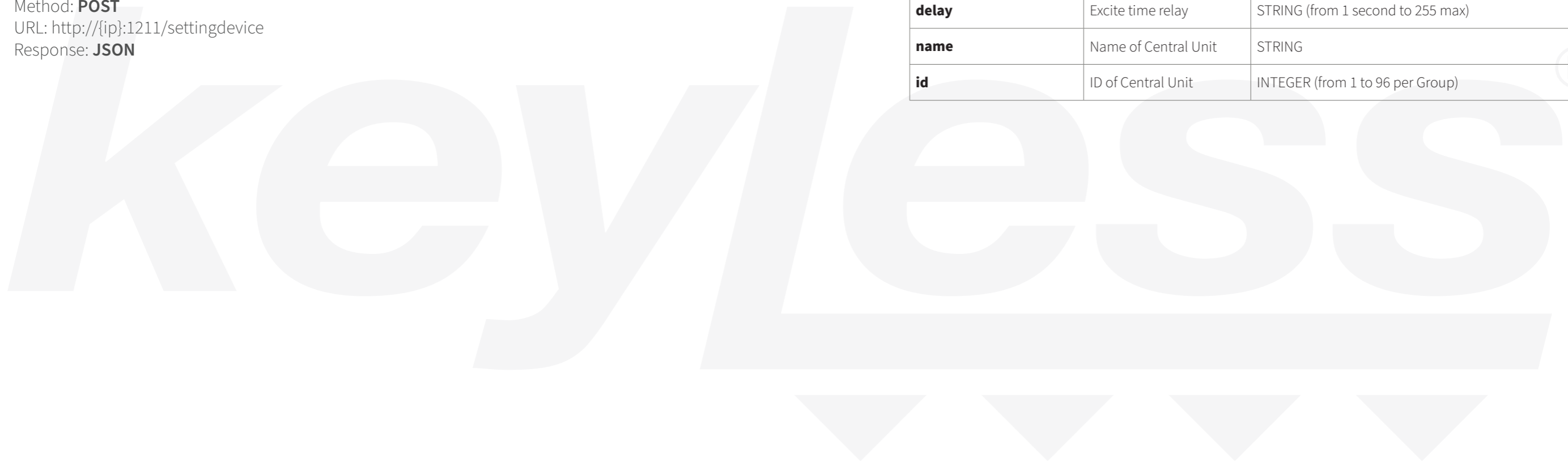
FUNCTION: CREATE OR MODIFY A CENTRAL UNIT

Method: **POST**
URL: `http://{ip}:1211/settingdevice`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	Password of the Central Unit	STRING (5 characters FIXED including 1 uppercase, 1 lowercase, 1 digit, 1 special character (* &% \$ # @))	Mandatory
profile	Type of Central unit settings	INTEGER (1 = Normal, 2 = Common)	Mandatory
groupname	NAME OF GROUP	STRING (12 characters FIXED including BLANK)	Mandatory
groupid	ID of Group	INTEGER	Mandatory
delay	Excite time relay	STRING (from 1 second to 255 max)	Mandatory
name	Name of Central Unit	STRING	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory



DELETE A CENTRAL UNIT



NOTICE



Pay attention: when a control unit is deleted, a slot is freed to generate a new control unit. Pay the utmost attention to assigning the new ID of the new control unit without duplicating an existing one

FUNCTION: DELETE A CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/deletedevice`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	NAME OF GROUP	STRING (12 characters FIXED including BLANK)	Mandatory
groupid	ID of Group	INTEGER	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory



LIST OF ALL CENTRAL UNIT



NOTICE



It queries the LOmnia Server to receive the list of all the control units present

FUNCTION: LIST OF ALL CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/device`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory



SET DATE & TIME



NOTICE



This QR Code sets the date and time of the single control unit.
The control units connected online (Internet) are automatically updated through the NTP Servers.

FUNCTION: SET DATE & TIME

Method: **POST**
URL: `http://{ip}:1211/settingdatetime`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (only 1 central unit)	Mandatory
day	Current Day	INTEGER (from 1 to 31)	Mandatory
month	Current Month	INTEGER (from 1 to 12)	Mandatory
year	Current Year	INTEGER (4 DIGIT)	Mandatory
hour	Current Hour	INTEGER (from 0 to 23)	Mandatory
minute	Current Minute	INTEGER(from 0 to 59)	Mandatory

Keyless

RESET BLACK LIST



NOTICE



This QR Code reset your entire black list on central unit

FUNCTION: RESET BLACK LIST

Method: **POST**

URL: `http://{ip}:1211/blacklist`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory



SET INTELRELAY OFF



NOTICE



This QR Code forces IntelRelay to shut down

FUNCTION: SET INTELRELAY OFF

Method: **POST**

URL: `http://{ip}:1211/intelrelay`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	GROUP NAME	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER	Mandatory



RESET KEYPAD MEMORY



NOTICE



By resetting the Keypad memory of the control unit, all blocked users can use their Code again (if valid in date and time).

In case, all the valid QR Code counters are reset

FUNCTION: RESET KEYPAD MEMORY

Method: **POST**

URL: `http://{ip}:1211/kepadreset`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory

Keyless®

LOCK OKULO PICO READER OR MHOM PICO KEYPAD



NOTICE



LOCK your Okulo PICO reader or bEar-MHOM PICO keypad .
The optical reader or the bEar of the PICO series, in case of theft or removal, even following a hardware reset, become unusable, because associated with your Account.

FUNCTION: LOCK YOUR OKULO PICO READER OR MHOM PICO KEYPAD

Method: **POST**

URL: `http://{ip}:1211/setidblock`

Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	STRING (from 1 to 96 per Group)	Mandatory

UNLOCK FOR OKULO PICO READER OR MHOM PICO KEYPAD



NOTICE



UNLOCK your Okulo PICO reader or MHOM PICO keypad

FUNCTION: UNLOCK YOUR OKULO PICO READER OR MHOM PICO KEYPAD

Method: **POST**

URL: `http://{ip}:1211/clearidblock`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	STRING (from 1 to 96 per Group)	Mandatory



CREATE A NEW QR CODE USER

NOTICE

Normal QR Code for Okulo version

In order to generate a Normal opening QR Code, it is necessary to send the array of control units (device) containing the control units that have a maximum of 2 different passwords and that are always of the same groupID and other mandatory parameters. (check list of variables)

NOTICE

Passepartout QR Code

the passepartout QR only evaluates the control unit password + all the parameters related to the dates. When it is created, it must contain at least one ID of the control unit series that have the same password.

The Passepartout QR Code can contain 2 IDs with 2 different passwords ... in that case, it will work on all control units including those 2 passwords. Logically, we made the group mandatory in truth the control unit does not evaluate it (only in the case of Passepartout) ... so it must be added but it is not necessary

NOTICE

Time slot

If your QR Code got time slot active and you want select a time slot for one entire day of week, store example
Friday all day access: **time1_5_from = 00:00** **time1_5_to = 23:59**

NOTICE

Nshot

If your QR Code contains nshots, you must also set a keypad code (always). Keypad Code can also be incremental. No hardware keypad is required.

FUNCTION: UNLOCK YOUR OKULO PICO READER OR MHOM PICO KEYPAD

Method: **POST**
URL: [http://\[ip\]:1211/api/create](http://[ip]:1211/api/create)
Response: **JSON**

VARIABLES

Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
devices[]	List of ID's of Central Unit	ARRAY(up to 96) (1 number each) In case of Passepartout (type = 2)min. 1 ID, max 2 ID's with different passwords	Mandatory
type	Type of QR Code	STRING (1 = Normal, 2 = Passepartout)	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
rele[]	List of Relays to excite	ARRAY (1 number each x 5)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited 1 = one shot 2 = nshots	Mandatory
nshot	quantity of shots	INTEGER (min. 2 - max. 65.000)	Mandatory if shot = 2
keypad	Access Numeric Keypad Code	INTEGER (5 numbers)	Optional, (Mandatory if nshot >= 2)
groupID	Central Unit Group only for Normal QR Code type	INTEGER (2 numbers)	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (X = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0= Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	Free field	STRING	Optional

LIST OF QR CODE USER



FUNCTION: LIST OF QR CODE USER

Method: **POST**
URL: http://{ip}:1211/api/list
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

Keyless[®]

BLOCK QR CODE USER



NOTICE



This special function it's compatible with both QR Code (Normal and Passepartout).
The resulting Block QR Code MUST be read from every Central Unit on which access is to be blocked (if Central Unit not connected only).

NOTICE



The idCode it's present on the RESPONSE of the NEW QR Code CREATION

FUNCTION: BLOCK QR CODE USER

Method: **POST**
URL: `http://{ip}:1211/api/delete`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	QR Code ID	INTEGER	Mandatory

LOmnia API

CHAPTER 2 LOGGER WI-FI



LOGGER WIFI



This chapter contain all instruction for Okulo, Okulo II, Okulo PICO, MHOM version.

INTRODUCTION

The API are reachable via REST API which return a JSON as a response.
All calls must be authenticated using your access TOKEN.

NOTICE



Manager RING Password & Logs for Logger Wi-Fi

The Manager RING Password & Logs for Logger Wi-Fi works only if the system is equipped with the optional Logger Wi-Fi for Okulo or MHOM Keypad
Once the RING Password or QR Code has been scanned or typed, the system logs the events
(all System events are logged too)

SET LOGGER WIFI



NOTICE



This QR Code sets the access data like(SSID & Password of the local Wi-fi) of the single control unit. The control units connected online (Internet) are automatically updated through the NTP Servers.

NOTICE



The Central Unit can be set through DIP Switch on board to change DST automatically on not. Restart the Central Unit after change

FUNCTION: SET LOGGER WIFI

Method: **POST**
URL: `http://{ip}:1211/setwifi`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupid	ID of Group	INTEGER	Mandatory
id	ID of Central Unit	INTEGER	Mandatory
ssid	SSID	STRING (max 32 characters)	Mandatory
password	Wi-Fi password	STRING	Optional
gmt	Time Zone (check the list below)	INTEGER	Mandatory
timerele	Time to hold relay wifi exciting	INTEGER (from 1 to 255 seconds)	Mandatory
ip	Wi-Fi IP Address	STRING	Optional
subnet	Wi-Fi subnet	STRING	Optional
routerip	Wi-Fi IP Router	STRING	Optional
dns1	Wi-Fi DNS 1	STRING	Optional
dns2	Wi-Fi DNS 1	STRING	Optional

GMT LIST

GMT	Value	Type	Request
GMT -11	89	INTEGER	Mandatory
GMT -10	90	INTEGER	Mandatory
GMT -9	91	INTEGER	Mandatory
GMT -8	92	INTEGER	Mandatory
GMT -7	93	INTEGER	Mandatory
GMT -6	94	INTEGER	Mandatory
GMT -5	95	INTEGER	Mandatory
GMT -4	96	INTEGER	Mandatory
GMT -3	97	INTEGER	Mandatory
GMT -2	98	INTEGER	Mandatory
GMT -1	99	INTEGER	Mandatory
GMT	100	INTEGER	Mandatory
GMT +1 (Europe Rome)	101	INTEGER	Mandatory
GMT +2	102	INTEGER	Mandatory
GMT +3	103	INTEGER	Mandatory
GMT +4	104	INTEGER	Mandatory
GMT +5	105	INTEGER	Mandatory
GMT +6	106	INTEGER	Mandatory
GMT +7	107	INTEGER	Mandatory
GMT +8	108	INTEGER	Mandatory
GMT +9	109	INTEGER	Mandatory
GMT +10	110	INTEGER	Mandatory
GMT +11	111	INTEGER	Mandatory

DELETE LOGGER WIFI



NOTICE



This API delete one Logger Wi-Fi from Database.
If you want to modify the Logger data on the Central Unit, you MUST reset the Logger on Central Unit and set a new Access data with a new Setup Wi-Fi QR Code

FUNCTION: DELETE LOGGER WIFI

Method: **POST**

URL: `http://{ip}:1211/deletewifi`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupid	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER	Mandatory





Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

FUNCTION: LIST OF ALL LOGGER WIFI

Method: **POST**

URL: http://{ip}:1211/listwifi

Response: **JSON**

Keyless[®]

CREATE A PERSONAL POST PATH FOR CENTRAL UNIT



NOTICE



It is essential to use this configuration to change the path of the LOGs and RING commands, diverting all the traffic of the control units to your own Server LOMnia

FUNCTION: CREATE A PERSONAL POST PATH FOR CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/setwebpath`

Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory
customwebserver	Path to Server	STRING (max 32 characters)	Mandatory

RESET PERSONAL POST PATH FOR CENTRAL UNIT



NOTICE



Back to original POST Path Okulo Keyless

FUNCTION: RESET PERSONAL POST PATH FOR CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/clearwebpath`

Response: **JSON**

Keyless®

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory

CREATE A NEW RING USER



NOTICE



Time slot

If your Password got time slot active and you want select a time slot for one entire day of week, store example

Friday all day access: **time1_5_from = 00:00** **time1_5_to = 23:59**

FUNCTION: CREATE A NEW RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-create`

Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	USER_PASSWORD	STRING	Mandatory
token_wifi	LOGGER_TOKEN	STRING	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited X = nshots	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	User ID	STRING	Optional



FUNCTION: EDIT RING USER

Method: **POST**
 URL: http://{ip}:1211/api/ring-edit
 Response: **JSON**



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	ID RING User code	INTEGER	Mandatory
password	USER_PASSWORD	STRING	Mandatory
token_wifi	LOGGER_TOKEN	STRING	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited X = nshots	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	User ID	STRING	Optional



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	ID RING User code	INTEGER	Mandatory

FUNCTION: DELETE RING USER

Method: **POST**
 URL: http://{ip}:1211/api/ring-delete
 Response: **JSON**



LIST OF RING USER



NOTICE



From date to date

If your call a List without **fromdate todate**, System return the last 6 month of users

FUNCTION: LIST OF RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-list`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
createdat	CREATION_DATE	DATE (YYYY-MM-DD)	Optional
fromdate	FROM_CREATION_DATE	DATE (YYYY-MM-DD)	Optional
todate	TO_CREATION_DATE	DATE (YYYY-MM-DD)	Optional



CHECK VALIDITY RING USER



NOTICE



This command check the validity of a single RING user

FUNCTION: CHECK VALIDITY RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-check`

Response: **JSON**

Keyless[®]

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	User Password	STRING	Mandatory

LIST OF LOG



NOTICE



30 days max

You can call a List of LOG within max 30 slot days

FUNCTION: CHECK VALIDITY RING USER

Method: **POST**

URL: `http://{ip}:1211/api/getlogger`

Response: **JSON**

Keyless[®]

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
wifi_token	Your WiFi Token	STRING	Mandatory
start	FROM_DATE	DATE (YYYY-MM-DD)	Mandatory
end	TO_DATE	DATE (YYYY-MM-DD)	Mandatory

CHECK CENTRAL UNIT RING CALL CONNECTION

NOTICE

UPDATE IN THIS REALESE



POST with acces_token only

return ALL Central Unit condition (connection)

REMEMBER: every single Central Unit call LOMnia every 5 seconds forever

If you check date & time call upper 10 seconds, probably Central Unit lost connection or power down

The Okulo PICO's return the level of the local Wi-Fi signal received. They also return the IP address with which they are connected

FUNCTION: CHECK CENTRAL UNIT RING CALL CONNECTION

Method: **POST**

URL: `http://{ip}:1211/ /api/wificonnect`

Response: **JSON**

VARIABLES

Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
wifi_token	Your WiFi Token	STRING	Optional



LOmnia API

CHAPTER 3 OKULO PLUS O+

OKULO PLUS



This chapter contain all instruction for Okulo PLUS version.

INTRODUCTION

The API are reachable via REST API which return a JSON as a response.
All calls must be authenticated using your access TOKEN.



CREATE A GROUP



NOTICE



Each group contain 1 Central Unit and this single can activate a maximum of 9.216 relays.

FUNCTION: CREATE A GROUP

Method: **POST**

URL: `http://{ip}:1211/groupsplus/create`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
name	Name of Group	STRING (12 characters Fixed including blanks)	Mandatory
note	Note of Group	STRING LONG TEXT	Optional



DELETE A GROUP



NOTICE



Pay attention to the group ID.
query the database to verify the ID of the group candidate for deletion

FUNCTION: CREATE A GROUP

Method: **POST**

URL: `http://{ip}:1211/groupsplus/delete`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
id	ID of Group	INTEGER	Mandatory

Keyless[®]



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

FUNCTION: LIST OF ALL GROUPS

Method: **POST**

URL: http://{ip}:1211/groupsplus

Response: **JSON**

Keyless®

CREATE A CENTRAL UNIT



NOTICE



Warning: the groupid it's the ID of the central Unit. Central Unit can be set with up to 768 relay each (in multiples of 8).
if you need to set several PLUS control units, each must be named differently with a different groupid.

FUNCTION: CREATE OR MODIFY A CENTRAL UNIT

Method: **POST**

URL: [http://\[ip\]:1211/settingdeviceplus](http://[ip]:1211/settingdeviceplus)

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	Password of the Central Unit	STRING (5 characters FIXED including 1 uppercase, 1 lowercase, 1 digit, 1 special character (* &% \$ # @))	Mandatory
profile	Type of Central unit settings	INTEGER (1 = Normal, 2 = Common)	Mandatory
groupname	NAME OF GROUP	STRING (12 characters FIXED including BLANK)	Mandatory
groupid	ID of Group	INTEGER	Mandatory
delay	Excite time relay	INTEGER (from 1 second to 255 max)	Mandatory
name	Name of Central Unit	STRING	Optional

Keyless

DELETE A CENTRAL UNIT



NOTICE



Pay attention: when a control unit is deleted, a slot is freed to generate a new control unit. Pay the utmost attention to assigning the new ID of the new control unit without duplicating an existing one

FUNCTION: DELETE A CENTRAL UNIT

Method: **POST**

URL: [http://\[ip\]:1211/deletedeviceplus](http://[ip]:1211/deletedeviceplus)

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	NAME OF GROUP	STRING (12 characters FIXED including BLANK)	Mandatory
groupid	ID of Group	INTEGER	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory



LIST OF ALL CENTRAL UNIT



NOTICE



It queries the LOmnia Server to receive the list of all the control units present

FUNCTION: LIST OF ALL CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/deviceplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

Keyless[®]

SET DATE & TIME



NOTICE



This QR Code sets the date and time of the single control unit.
The control units connected online (Internet) are automatically updated through the NTP Servers.

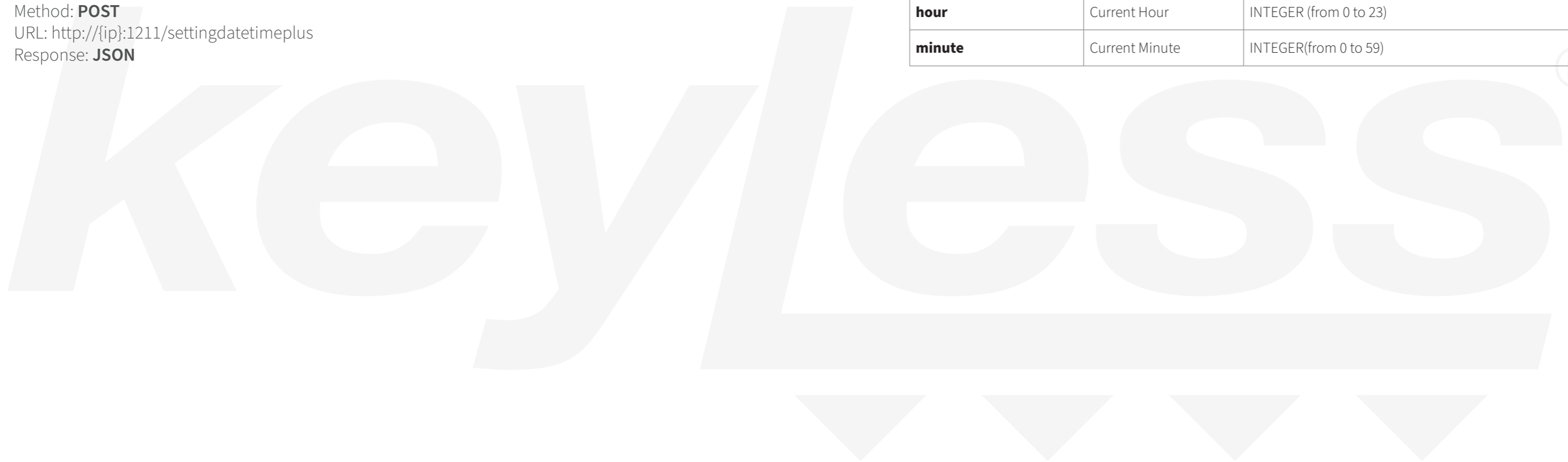
FUNCTION: SET DATE & TIME

Method: **POST**
URL: `http://{ip}:1211/settingdatet imeplus`
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
groupid	ID of Central Unit	INTEGER (only one central unit)	Mandatory
day	Current Day	INTEGER (from 1 to 31)	Mandatory
month	Current Month	INTEGER (from 1 to 12)	Mandatory
year	Current Year	INTEGER (4 DIGIT)	Mandatory
hour	Current Hour	INTEGER (from 0 to 23)	Mandatory
minute	Current Minute	INTEGER(from 0 to 59)	Mandatory



RESET BLACK LIST



NOTICE



This QR Code reset your entire black list on central unit

FUNCTION: RESET BLACK LIST

Method: **POST**

URL: `http://{ip}:1211/blacklistplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory
groupid	ID of Group	INTEGER	Mandatory



SET INTELRELAY OFF



NOTICE



This QR Code forces IntelRelay to shut down

FUNCTION: SET INTELRELAY OFF

Method: **POST**

URL: `http://{ip}:1211/intelrelayplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	GROUP NAME	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER	Mandatory



CREATE A NEW QR CODE USER



NOTICE



Normal QR Code for Okulo PLUS version

In order to generate a Normal opening QR Code, it is necessary to send the array of control units (device) containing the control units that have a maximum of 2 different passwords and that are always of the same groupID and other mandatory parameters. (check list of variables)

NOTICE



Passepartout QR Code

the passepartout QR only evaluates the control unit password + all the parameters related to the dates. When it is created, it must contain at least one ID of the control unit series that have the same password.

The Passepartout QR Code can contain 2 IDs with 2 different passwords ... in that case, it will work on all control units including those 2 passwords
Logically, we made the group mandatory in truth the control unit does not evaluate it (only in the case of Passepartout) ... so it must be added but it is not necessary

NOTICE



Time slot

If your QR Code got time slot active and you want select a time slot for one entire day of week, store example
Friday all day access: **time1_5_from = 00:00** **time1_5_to = 23:59**

NOTICE



Nshot

If your QR Code contains nshots, you must also set a keypad code (always). Keypad Code can also be incremental. No hardware keypad is required.

FUNCTION: CREATE A NEW QR CODE USER

Method: **POST**
URL: [http://\[ip\]:1211/api/createplus](http://[ip]:1211/api/createplus)
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
type	Type of QR Code	STRING (1 = Normal, 2 = Passepartout)	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
rele	Number of Relay to excite (you can excite one relay ONLY)	INTEGER if type = 1 (1 to 768 = direct relay) if type = 2 (0 = type on keypad number of relay to excite + #)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited 1 = one shot 2 = nshots	Mandatory
nshot	quantity of shots	INTEGER (min. 2 - max. 65.000)	Mandatory if shot = 2
groupID	Central Unit Group only for Normal QR Code type	INTEGER (2 numbers)	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (X = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0= Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	Free field	STRING	Optional

LIST OF QR CODE USER



FUNCTION: LIST OF QR CODE USER

Method: **POST**

URL: `http://{ip}:1211/api/listplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

Keyless[®]

BLOCK QR CODE USER



NOTICE



This special function it's compatible with both QR Code (Normal and Passepartout).

The resulting Block QR Code MUST be read from every Central Unit on which access is to be blocked (if Central Unit not connected only).

NOTICE



The idCode it's present on the RESPONSE of the NEW QR Code CREATION

FUNCTION: BLOCK QR CODE USER

Method: **POST**

URL: `http://{ip}:1211/api/deleteplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	QR Code ID	INTEGER	Mandatory

LOmnia API

CHAPTER 4 LOGGER WI-FI PLUS



LOGGER WIFI



This chapter contain all instruction for Okulo, Okulo II, OKULO PLUS, Okulo PICO, MHOM version.

INTRODUCTION

The API are reachable via REST API which return a JSON as a response.
All calls must be authenticated using your access TOKEN.

NOTICE



Manager RING Password & Logs for Logger Wi-Fi

The Manager RING Password & Logs for Logger Wi-Fi works only if the system is equipped with the optional Logger Wi-Fi for Okulo, OKULO PLUS or MHOM Keypad
Once the RING Password or QR Code has been scanned or typed, the system logs the events
(all System events are logged too)

SET LOGGER WIFI



NOTICE



This QR Code sets the access data like(SSID & Password of the local Wi-fi) of the single control unit. The control units connected online (Internet) are automatically updated through the NTP Servers.

NOTICE



The Central Unit can be set through DIP Switch on board to change DST automatically on not. Restart the Central Unit after change

FUNCTION: SET LOGGER WIFI

Method: **POST**
URL: [http://\[ip\]:1211/setwifiplus](http://[ip]:1211/setwifiplus)
Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupid	ID of Group	INTEGER	Mandatory
id	ID of Central Unit	INTEGER	Mandatory
groupname	Name of Group	STRING	Mandatory
ssid	SSID	STRING (max 32 characters)	Mandatory
password	Wi-Fi password	STRING	Optional
gmt	Time Zone (check the list below)	INTEGER	Mandatory
timerele	Time to hold relay wifi exciting	INTEGER (from 1 to 255 seconds)	Mandatory
ip	Wi-Fi IP Address	STRING	Optional
subnet	Wi-Fi subnet	STRING	Optional
routerip	Wi-Fi IP Router	STRING	Optional
dns1	Wi-Fi DNS 1	STRING	Optional
dns2	Wi-Fi DNS 1	STRING	Optional

GMT LIST

GMT	Value	Type	Request
GMT -11	89	INTEGER	Mandatory
GMT -10	90	INTEGER	Mandatory
GMT -9	91	INTEGER	Mandatory
GMT -8	92	INTEGER	Mandatory
GMT -7	93	INTEGER	Mandatory
GMT -6	94	INTEGER	Mandatory
GMT -5	95	INTEGER	Mandatory
GMT -4	96	INTEGER	Mandatory
GMT -3	97	INTEGER	Mandatory
GMT -2	98	INTEGER	Mandatory
GMT -1	99	INTEGER	Mandatory
GMT	100	INTEGER	Mandatory
GMT +1 (Europe Rome)	101	INTEGER	Mandatory
GMT +2	102	INTEGER	Mandatory
GMT +3	103	INTEGER	Mandatory
GMT +4	104	INTEGER	Mandatory
GMT +5	105	INTEGER	Mandatory
GMT +6	106	INTEGER	Mandatory
GMT +7	107	INTEGER	Mandatory
GMT +8	108	INTEGER	Mandatory
GMT +9	109	INTEGER	Mandatory
GMT +10	110	INTEGER	Mandatory
GMT +11	111	INTEGER	Mandatory

DELETE LOGGER WIFI



NOTICE



This API delete one Logger Wi-Fi from Database.
If you want to modify the Logger data on the Central Unit, you MUST reset the Logger on Central Unit and set a new Access data with a new Setup Wi-Fi QR Code

FUNCTION: DELETE LOGGER WIFI

Method: **POST**
URL: [http://\[ip\]:1211/deletewifiplus](http://[ip]:1211/deletewifiplus)
Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupid	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER	Mandatory



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory

FUNCTION: LIST OF ALL LOGGER WIFI

Method: **POST**

URL: http://{ip}:1211/listwifiplus

Response: **JSON**

Keyless[®]

CREATE A PERSONAL POST PATH FOR CENTRAL UNIT



NOTICE



It is essential to use this configuration to change the path of the LOGs and RING commands, diverting all the traffic of the control units to your own Server LOMnia

FUNCTION: CREATE A PERSONAL POST PATH FOR CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/setwebpathplus`

Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory
groupid	ID of the Group	INTEGER	Mandatory
customwebserver	entire Path to Server	STRING (max 32 characters) (<code>http://.....:1211</code>)	Mandatory

RESET PERSONAL POST PATH FOR CENTRAL UNIT



NOTICE



Back to original POST Path Okulo Keyless

FUNCTION: RESET PERSONAL POST PATH FOR CENTRAL UNIT

Method: **POST**

URL: `http://{ip}:1211/clearwebpathplus`

Response: **JSON**

Keyless[®]

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
groupname	Name of Group	STRING (12 characters including blanks)	Mandatory
id	ID of Central Unit	INTEGER (from 1 to 96 per Group)	Mandatory
groupid	ID of the Group	INTEGER	Mandatory

CREATE A NEW RING USER



NOTICE



Time slot

If your Password got time slot active and you want select a time slot for one entire day of week, store example

Friday all day access: **time1_5_from = 00:00** **time1_5_to = 23:59**

FUNCTION: CREATE A NEW RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-createplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	USER_PASSWORD	STRING	Mandatory
token_wifi	LOGGER_TOKEN	STRING	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited X = nshots	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	User ID	STRING	Optional



FUNCTION: EDIT RING USER

Method: **POST**
 URL: http://{ip}:1211/api/ring-editplus
 Response: **JSON**



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	ID RING User code	INTEGER	Mandatory
password	USER_PASSWORD	STRING	Mandatory
token_wifi	LOGGER_TOKEN	STRING	Mandatory
startDay	Beginning Day of validity	STRING (2 characters)	Mandatory
startMonth	Beginning Month of validity	STRING (2 characters)	Mandatory
startYear	Beginning Year of validity	STRING (2 characters)	Mandatory
startHour	Beginning Hour of validity	STRING (2 characters)	Mandatory
startMinute	Beginning Minute of validity	STRING (2 characters)	Mandatory
endDay	End Day of validity	STRING (2 characters)	Mandatory
endMonth	End Month of validity	STRING (2 characters)	Mandatory
endYear	End Year of validity	STRING (2 characters)	Mandatory
endHour	End Hour of validity	STRING (2 characters)	Mandatory
endMinute	End Minute of validity	STRING (2 characters)	Mandatory
shot	N SHOTS	INTEGER 0 = unlimited X = nshots	Mandatory
time1_X_from	Beginning 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time1_X_to	End 1st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time1_X_from = FULL)
time2_X_from	Beginning 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Optional
time2_X_to	End 2st time slot of day	STRING (0 = Sunday, Saturday = 6) (time = HH:mm)	Mandatory if (time2_X_from = FULL)
name	User name	STRING	Optional
surname	User surname	STRING	Optional
email	User email	STRING	Optional
phone	User phone	STRING	Optional
idutente	User ID	STRING	Optional

DELETE A RING USER



FUNCTION: DELETE RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-deleteplus`

Response: **JSON**



VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
idCode	ID RING User code	INTEGER	Mandatory

LIST OF RING USER



NOTICE



From date to date

If your call a List without **fromdate todate**, System return the last 6 month of users

FUNCTION: LIST OF RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-listplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
createdat	CREATION_DATE	DATE (YYYY-MM-DD)	Optional
fromdate	FROM_CREATION_DATE	DATE (YYYY-MM-DD)	Optional
todate	TO_CREATION_DATE	DATE (YYYY-MM-DD)	Optional



CHECK VALIDITY RING USER



NOTICE



This command check the validity of a single RING user

FUNCTION: CHECK VALIDITY RING USER

Method: **POST**

URL: `http://{ip}:1211/api/ring-checkplus`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
password	User Password	STRING	Mandatory



LIST OF LOG



NOTICE



30 days max

You can call a List of LOG within max 30 slot days

FUNCTION: CHECK VALIDITY RING USER

Method: **POST**

URL: `http://{ip}:1211/api/getlogger`

Response: **JSON**

VARIABLES



Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
wifi_token	Your WiFi Token	STRING	Mandatory
start	FROM_DATE	DATE (YYYY-MM-DD)	Mandatory
end	TO_DATE	DATE (YYYY-MM-DD)	Mandatory



CHECK CENTRAL UNIT RING CALL CONNECTION

NOTICE



POST with access_token only

return ALL Central Unit condition (connection)

REMEMBER: every single Central Unit call LOMnia every 5 seconds forever

If you check date & time call upper 10 seconds, probably Central Unit lost connection or power down

FUNCTION: CHECK CENTRAL UNIT RING CALL CONNECTION

Method: **POST**

URL: `http://{ip}:1211/ /api/wificonnect`

Response: **JSON**

VARIABLES

Variable	Value	Type	Request
access_token	Your API Token	STRING	Mandatory
wifi_token	Your WiFi Token	STRING	Optional

Keyless®



Despite the accuracy and attention paid to the creation, updating and integration of the contents and information contained in this document, these contents and information may contain printing errors, inaccuracies or inconsistencies in relation to which Keyless® does not provide no guarantee - express or tacit - even by way of mere expectation or expectation. In particular, by way of example and without limitation, no insurance or guarantee is provided regarding their completeness, completeness and compliance with the current organization, company structure or type, characteristics, marketability or suitability for the purpose of the products, services and Keyless® activity, which is available to provide all the necessary information to the interested parties. Keyless® reserves the right to make changes to the information contained in this document without notice, as well as to change and modify the products, services and activities described therein without this giving rise to rights of any kind for third parties. It is important to keep this document in order to be able to consult it at all times. In the interest of improving its internal design, operational functionality and / or reliability, Keyless® reserves the right to make changes to the products and software commands described in this document without any prior notice or communication in written or electronic form. Keyless® declines all responsibility during the use or application of the products or commands described. All the brands and logos shown in this manual are respectively registered and reserved by the individual companies that hold their rights. For any controversy, the Court of Treviso - Italy is now elected.