

LCI Web API v1 规范

[目的](#)

[步骤概要](#)[步骤详细入](#)

[口点 API 详细:共同事](#)

[项](#)

[API 细节:访问 键和 签名 键生成并更新访问 键和 签名](#)

[键删除访问 键和 签名 键](#) [API 细节:会话](#)

[会话的生成](#)

[会话的删除](#)

[API 详情:电梯](#)

[电梯使用登记](#)

[电梯的使用解除](#)

[电梯目的地指定\(呼叫\)](#)

[电梯到达和门的开关状态的确认](#)

[要求继续开门，上车完成，上车中止，下车完成，下车中止](#) [API 详细:自动门安全门](#)

[自动门、安全门的使用登记](#)

[自动门、安全门的使用解除](#)

[自动门、安全门的开门、开锁要求](#)

[自动门、安全门的开关状态的确认](#)

[更新日志](#)

目的

制定用于在 HTTPS 中使用 LCI 协议的 Web API 的规范。

※注意:API 服务器是代替 MQTTS 客户端的结构，因此延迟增加。事先一定要验证实用上是否没问题。

步骤概要

- 1.操作员利用作为 LCI 机器人账户分发的私有密钥从 API 服务器获取访问密钥和签名密钥。
- 2.操作员在机器人(运行管理服务器也可以)上配置取得的钥匙。
- 3.机器人使用访问密钥和签名密钥在 API 服务器上生成 LCI 客户端的会话，并获取会话 ID。
- 4.机器人使用获取的会话 ID 通过 API 服务器利用 LCI 协议。

详细步骤

- 1.客户端 POST 到/key，请求访问密钥和签名密钥。

a.在 POST 时的 body 中，以 json 形式加入以下参数。

- mqtt_server: mqtt 服务器的 URL。普通 lci.octa8.link
- robot_id: LCI 机器人账号的 id。ClientID 文件中的字符串
- api:
 - 生成/更新访问键时使用“Create”
 - 删除访问键时使用“Destroy”
- token:代币

- b.令牌是 JWT 形式的字符串，使用机器人账户的私钥在 RS256 签名。

JWT 的有效载荷如下

- iss:用“Octa Robotics”固定
- sub: MQTT 服务器的 URL。普通 lci.octa8.link
- aud: LCI 机器人账号的 ID。ClientID 文件内的字符串
- exp:希望的有
效期限(纪元秒)
- iat:要求的时间(纪元秒)

○有 Salt 的作用，如果过去超过 180 秒，就会被拒绝。 c.

ResponseCode 200 的时候，以下以 json 形式进入 body

- access_key:访问键(字符串)。用于生成/删除会话
- sign_key:签名用键(字符串)。用于生成/删除会话。
- expiration_time:从系统分配的 access_key 和 sign_key 的有
有效期限(纪元秒)

- 2.客户端 POST 到/session，并在利用 MQTTS 会话的 a. POST 时的 body 中

以 json 形式加入以下参数

- mqtt_server 支持
- robot_id
- api:后述

■ token:代币

●仅在生成/删除会话时必需 ■其他, api 所需的参数(后述)

b.令牌是使用 sign_key 在 HS256 上签名的 JWT 格式的字符串

JWT 的有效载荷如下

- iss:用“Octa Robotics”固定
- sub: MQTT 服务器的 URL
- aud: LCI 机器人账号的 ID
- exp:从系统分配的 access_key 和 sign_key 的有效期限(纪元秒)
- iat:要求的时间(纪元秒)

○有 Salt 的作用, 如果过去超过 180 秒, 就会被拒绝。 ●access_key:从 API 服务器发出的访问密钥 c. ResponseCode 200 时, 以下以 json 形式进入 body

当 api = “Create” 时, 生成 MQTT 会话, 返回如下 ●session_id:会话 id。在以后的交换中作为参数使用。

- 会话在客户端访问中断 10 分钟时自动结束。
- 在建立新的会话的情况下, 旧的会话结束。

■api == “Destroy” 时, 结束指定的 session_id 的会话。

■其他 API 时, 根据 lci01 - speci -002 LCI 协议说明书的 API, 从 LCI 直接返回 json 形式的响应。

- 在参数中加入 MQTT 的 topic 中所需的信息(后述) ●在 180 秒内没有来自 LCI 的响应时, 回复 REQUEST_timeout

入口点

<https://api.lci.octa8.link/v1>

API 详情:共同事项

●所有,

○Method 是 POST

Body 发送和回复都是 Content-Type: application/json

API 细节:访问键和签名键

生成和更新访问键和签名键

path	/key
method	post
api	create
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa 8 . link” , “robot _ id” : “oct00000” 、 “api” : “create” 、 “token” :< jwt ></pre>
< jwt > header	<pre>{ “typ” : “JWT” , “alg” : “RS256” }</pre>
< jwt > payload	<pre>iss: “Octa Robotics” , { sub: “lci.octa8.link” , aud: “OCT00000” , exp: 1758073069,165820514 iat: 3 }</pre>
signature key <JWT>	Private key of a 机器人 account provided by Octa Robotics
Body example (response) when “200ok”	<pre>{ “access key” : “xxxxxxxxxxxxxxxxxxxx” , “sign _ key” : “yyyyyyyyyyyyyyyyyyyy” 、 “expiration _ time” :1758073069 }</pre>
五十	

删除访问键和签名键

path	/key
method	post
api	destroy
身体反射(request)	<pre>; “mqtt_server 支持” : “lci.octa 8 . link” , “robot _ id” : “oct00000” 、 “api” : “destroy” 、 “token” :< jwt > ,</pre>

	<pre>“access key” : “xxxxxxxxxxxxxxxx” }</pre>
< jwt > header	<pre>{ “typ” : “JWT” , “alg” : “RS256” }</pre>
< jwt > payload	<pre>iss: “Octa Robotics” , sub: “lci.octa8.link” , aud: “OCT00000” , exp: 1758073069,1658206143 iat:</pre>
signature key <JWT>	Private key of a 机器人 account provided by Octa Robotics
Body example (response) when “200ok”	<pre>{ “access key” : “xxxxxxxxxxxxxxxx” }</pre>
五十	

API 详情:会话

会话的生成

path	/ session
method	post
ap i	create
身体反射(request)	<pre>“mqtt _ server 支持” : “lci.octa 8 . link” , “robot _ id” : “oct00000” 、 “api” : “create” 、 “token” :< jwt > }</pre>
< jwt > header	<pre>{ “typ” : “JWT” } “alg” : “HS256”</pre>
< jwt > payload	<pre>; “iss” : “Octa 机器人” “sub” : “lci.octa 8 . link” , “aud” : “oct00000” , “exp” :1758073069, “iat” :1658207143, “access key” : “xxxxxxxxxxxxxxxx”</pre>

}	
signature key <JWT>	sign_key obtained via /key
Body example (response) when “200ok”	“session_id” : “zzzzzzzzzzzzzzzzzzzz” }

五十

会话的删除

path	/ session
method	post
api	destroy
身体反射(request)	“mqtt_server” : “lci.octa8.link” , { “robot_id” : “OCT00000” , “api” : “Destroy” , “token” :<JWT>, “session_id” : “zzzzzzzzzzzzzzzzzzzz” }
< jwt > header	{ “typ” : “JWT” , “alg” : “HS256”
< jwt > payload	“iss” : “Octa 机器人” { “sub” : “lci.octa 8 . link” , “aud” : “oct00000” , “exp” :1758073069, “iat” :1658208143, “access key” : “xxxxxxxxxxxxxxxxxx”
signature key <JWT>	sign_key obtained via /key
Body example (response) when “200ok”	{ “session id” : “zzzzzzzzzzzzzzzzzzzz” }
五十	

API 详情:电梯

电梯使用登记

path	/ .
------	-----

method	post
api	registration
身体反射(request)	<pre>“mqtt_server 支持”：“lci.octa 8.link”， { “robot_id”：“oct00000”， “api”：“registration”， “session_id”：“zzzzzzzzzzzzzzzzzz”， “bldg_id”：“simulator”， “bank_id”：“3”， “elevator_id”：“1”， “timestamp”：1658208153.125</pre>
Body example (response) when “200ok”	<pre>“result”：1，“elevator_id”：“1”， { “timestamp”：1658208173.125， “requestd_robot_id”：“OCT00000”， “requestd_timestamp”：1658208153.125</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。 Response body 是 RegistrationResult 的有效载荷。

电梯的使用解除

path	/session
method	post
api	release
身体反射(request)	<pre>“mqtt_server 支持”：“lci.octa 8.link”， { “robot_id”：“oct00000”， “api”：“release”， “session_id”：“zzzzzzzzzzzzzzzzzz”， “bldg_id”：“simulator”， “bank_id”：“3”， “elevator_id”：“1”， “timestamp”：1658209253.125</pre>

Body example (response) when “200ok”	<pre>{ “result” :1, “timestamp” :1658208273.125, “requestd_robot_id” : “OCT00000” , “requestd_timestamp” :1658209253.125 }</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。

有道文档翻译
pdf.youdao.com

Response body 是 ReleaseResult 的有效载荷。

电梯目的地指定(呼叫)

path	/session
method	post
api	callelevator
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa8.link” , “robot_id” : “oct00000” , {} “api” : “callelevator” , “session_id” : “zzzzzzzzzzzzzzzzzz” , “bldg_id” : “simulator” , “bank_id” : “3” , “elevator_id” : “1” , “timestamp” :1658208353.125, “origination” : “1f” , “origination_door” :1, “distination” : “2f” , “distination_door” :1</pre>
Body example (response) when “200ok”	<pre>“result” :1, {} “timestamp” :1658208363.125, “requestd_robot_id” : “OCT00000” , “requestd_timestamp” :1658208353.125</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。 Response body 是 CallElevatorResult 的有效载荷。

电梯到达和门的开关状态的确认

path	/session
method	post
api	requestelevatorstatus
身体反射(request)	;

```
“mqtt_server” : “lci.octa8.link” ,  
“robot_id” : “OCT00000” ,  
“api” : “RequestElevatorStatus” ,  
“session_id” : “zzzzzzzzzzzz” ,  
“bldg_id” : “simulator” ,  
“bank_id” : “3” ,  
  
“elevator_  
id” : “1” ,  
  
“timestamp” :1658208453.125
```

有道文档翻译
pdf.youdao.com

	}
身体反射(response) when “200ok”	; “result” :1, “floor” : “1 f” , “door” :1, “direction” :3, } “timestamp” :1658208563.125, “requested_robot译 文:id” : “oct00000” , “requested timestamp” :1658208453.125
五十	各参数详见 lci01 - speci -002 LCI 协议 参照样书。 Response body 是 ElevatorStatus 的有效载荷 有。

要求继续打开车门，上车完成，上车中止，下车完成，下车中止

path	/ session
method	post
api	robotstatus
身体反射(request)	“mqtt_server 支持” : “lci.octa 8.link” , { “robot_id” : “oct00000” , “api” : “robotstatus” , “session id” : “zzzzzzzzzzzzzzzzzz” , “bldg_ id” : “simulator” , “bank_ id” : “3” , “elevator_ id” : “1” , “timestamp” :1658208453.125, “state” :5
Body example (response) when “200ok”	{ “result” :1, “timestamp” :1658208563.125, “requestd_robot_id” : “OCT00000” , “requestd_timestamp” :1658208453.125
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明 书。 Response body 是 RobotStatusResult 的有效载荷。

API 详情:自动门安全门

自动门、安全门的使用登记

path / session

有道文档翻译
pdf.youdao.com

method	post
api	registration
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa 8 . link” , { “robot _id” : “oct00000” , “api” : “registration” , “session id” : “zzzzzzzzzzzzzzzzzzzz” , “bldg _ id” : “simulator” , “floor _ id” : “3 f” , “door id” : “1” , “timestamp” :1658208153.125</pre>
Body example (response) when “200ok”	<pre>“result” :1 , { “timestamp” :1658208173.125 , “requestd_robot _id” : “OCT00000” , “requestd_timestamp” :1658208153.125</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。 Response body 是 RegistrationResult 的有效载荷。

自动门、安全门的使用解除

path	/session
method	post
api	release
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa 8 . link” , { “robot _id” : “oct00000” , “api” : “release” , “session id” : “zzzzzzzzzzzzzzzzzzzz” , “bldg _ id” : “simulator” , “floor _ id” : “3 f” , “door id” : “1” , “timestamp” :1658209253.125</pre>

Body example (response) when “200ok”	<pre>{ “result” :1, “timestamp” :1658208273.125, “requestd_robot_id” : “OCT00000” , “requestd_timestamp” :1658209253.125 }</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。 Response body 是 ReleaseResult 的有效载荷

	看。
--	----

自动门、安全门的开门、开锁要求

path	/ session
method	post
api	opendoor
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa8 . link” , { “robot _id” : “oct00000” , “api” : “opendoor” , “session id” : “zzzzzzzzzzzzzzzzzz” , “bldg _id” : “simulator” , “floor _id” : “3 f” , “door id” : “1” , “timestamp” :1658208353.125,</pre>
Body example (response) when “200ok”	<pre>“result” :1, { “timestamp” :1658208363.125, “requestd_robot _id” : “OCT00000” , “requestd_timestamp” :1658208353.125</pre>
五十	各参数的详细内容请参考 lci01 - speci -002 LCI 协议说明书。 Response body 是 OpenDoorResult 的有效载荷。

自动门、安全门的开关状态的确认

path	/ session
method	post
api	requestdoorstatus
身体反射(request)	<pre>“mqtt_server 支持” : “lci.octa8 . link” , { “robot _id” : “oct00000” , “api” : “requestdoorstatus” , “session id” : “zzzzzzzzzzzzzzzzzz” , “bldg _id” : “simulator” ,</pre>

	<pre>“floor _ id” : “3 f” , “door id” : “1” , “timestamp” :1658208453.125</pre>
Body example (response) when “200ok”	<pre>; “result” :1, “door” :1,</pre>

	<div>“timestamp” :1658208563.125,</div> <div>“requested robot 译文:id” : “oct00000” ,</div> <div>}</div> <div>“requested timestamp” :1658208453.125</div>
五十	<div>各参数详见 lci01 - speci -002 LCI 协议参照样书。</div> <div>Response body 是 DoorStatus 的有效载荷。</div>

更新日志

版	变更内容
2022-07-19	初版发行