

ChirpStack 下发数据到节点

注：本文档适用于 ChirpStack3.x 版本。

目录

ChirpStack 下发数据到节点.....	1
1 概述.....	4
2 OTAA CLASS A.....	4
2.1 创建 Otaa class A 的 Device-profile.....	4
2.2 创建 Otaa class A 的 Device.....	6
2.3 设置节点信息.....	8
2.4 服务器下发数据到节点.....	9
3 OTAA CLASS C.....	10
3.1 创建 Otaa class C 的 Device-profile.....	10
3.2 创建 Otaa class C 的 Device.....	12
3.3 设置节点信息.....	13
3.4 服务器下发数据到节点.....	13
4 ABP CLASS A.....	14
4.1 创建 ABP class A 的 Device-profile.....	14
4.2 创建 ABP class A 的 Device.....	15
4.3 设置节点信息.....	18
4.4 服务器下发数据到节点.....	19
5 ABP CLASS C.....	20
5.1 创建 ABP class C 的 Device-profile.....	20
5.2 创建 ABP class C 的 Device.....	22
5.3 设置节点信息.....	22

5.4 服务器下发数据到节点.....	23
6 附录一：使用中可能遇到的问题.....	24

1 概述

该文档将分别介绍 OTAA CLASS A、OTAA CLASS C、ABP CLASS A 和 ABP CLASS C 模式下的数据下发。

本章节仅适用于 RAK 的网关、RAK 树莓派网关内置的 ChirpStack 服务器、RAK 提供的云 ChirpStack 服务器以及 RAK 的节点 (RAK4200、RAK811、RAK7204、RAK7205、RAK5205 等)。

2 OTAA CLASS A

2.1 创建 Otaa class A 的 Device-profile

首先我们创建 Otaa class A 的 Device-profile, 该 Device-profile 只需要配置以下两页内容:



Device-profiles / dev-profile_otaa_class_a

GENERAL

JOIN (OTAA / ABP)

CLASS-B

CLA

Device-profile name *

dev-profile_otaa_class_a

A name to identify the device-profile.

LoRaWAN MAC version *

1.0.2

The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision *

A

Revision of the Regional Parameters specification supported by the device.

Max EIRP *

0

Maximum EIRP supported by the device.

Geolocation buffer TTL (seconds)

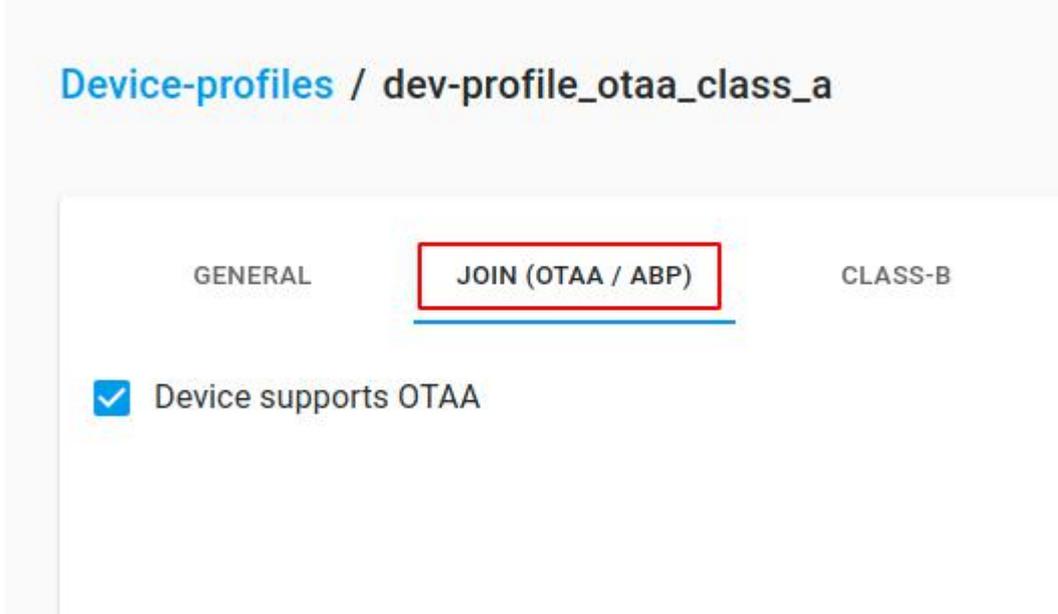
0

The time in seconds that historical uplinks will be stored in the geolocation buffer.

Geolocation minimum buffer size

0

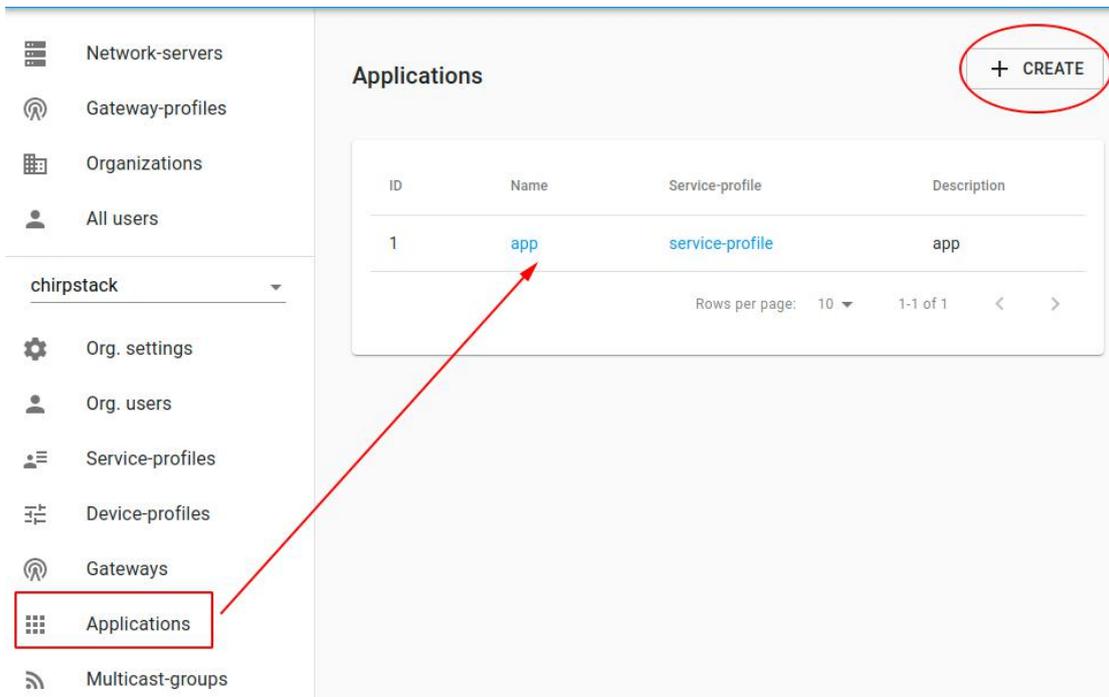
The minimum buffer size required before using geolocation (when enabled in the Service Profile).¹



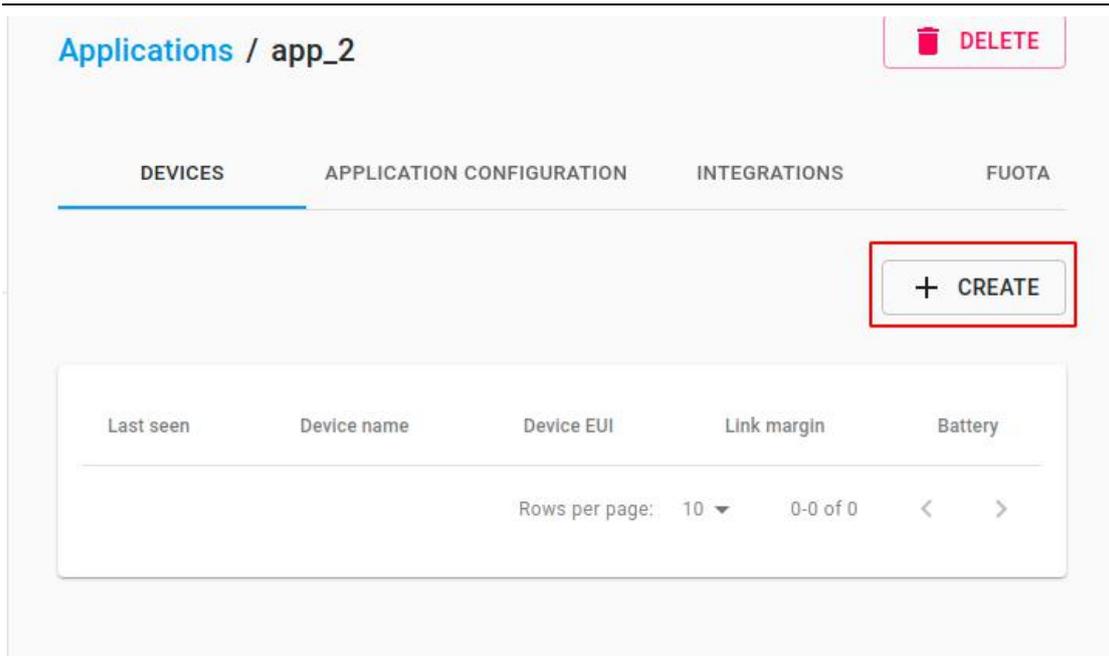
2.2 创建 Otaa class A 的 Device

创建完成 Device-profile 之后，接下来我们创建一个 otaa class a 的节点。

- 1) 点击 Applications，打开一个应用。如果没有应用，请点击右上角的 CREATE 按钮自行创建有一个 application。



- 2) 在打开的 application 中，点击右上角的 CREATE，创建一个节点



- 3) 如下图所示依次填入相应内容。Device EUI 我们可以自行填写，也可以点击按钮随机生成。Device-profile 中我们必须选择刚刚创建的 otaa class a 的 device-profile。



- 4) 填写完相应内容之后，点击右下角的 CREATE DEVICE，跳转到生成 Application Key 的

页面，点击按钮可生成随机的 Application key，此处需要用户保存该 Application Key 以便后续使用：



2.3 设置节点信息

使用上一步生成的 device EUI 和 Application key 设置我们节点的三元组信息。对于 ChirpStack 服务器，节点只需要设置 device EUI 和 Application key 即可，对于 app EUI 无需设置，但请不要将 app EUI 设置为全 0。

```
at+set_config=lora:dev_eui:xxxxxxxxxxxxxxxx
```

```
at+set_config=lora:app_eui:0000000000000555
```

```
at+set_config=lora:app_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:class:0
```

```
at+set_config=lora:join_mode:0
```

```
at+set_config=lora:work_mode:0
```

以上 at 指令设置了节点的三元组信息，以及将节点设置为 otaa class a 模式。

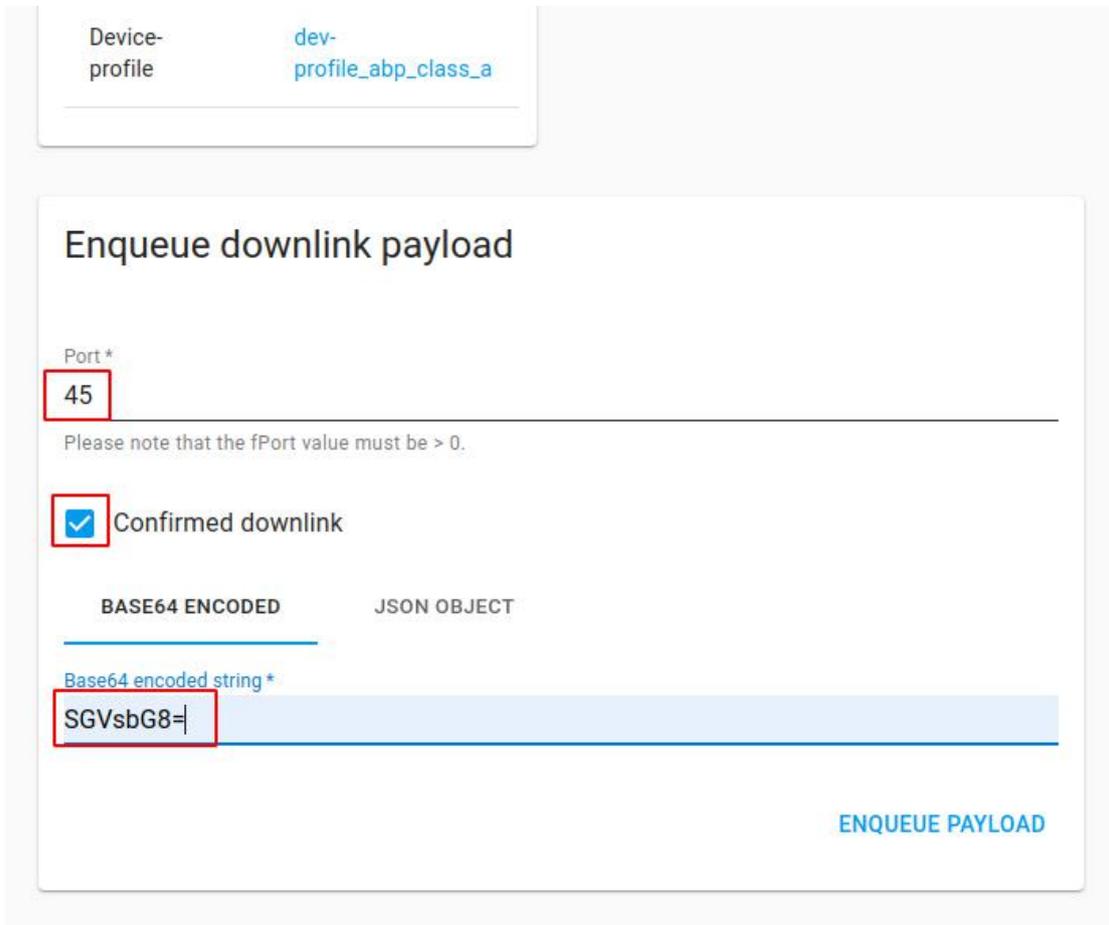
设置成功之后，可以使用 `at+get_config=lora:status` 检查是否设置成功。

2.4 服务器下发数据到节点

- 1) 节点需要先 join 成功后，再成功发送一条数据到服务器。

```
[15:23:57.041]发→◇at+join
□
[15:24:02.237]收←◆OK Join Success
[15:24:39.240]发→◇at+send=lora:1:48656c6c66f
□
[15:24:40.352]收←◆OK
at+recv=0,-96,7,0
```

- 2) 在 ChirpStack 页面中点开我们刚刚创建的 Device。Port 取值范围是 1-255, Confirmed downlink 是可选项，“SGVsbG8=” 是 “Hello” 的 base64 编码



Device-profile [dev-profile_abp_class_a](#)

Enqueue downlink payload

Port *

Please note that the fPort value must be > 0.

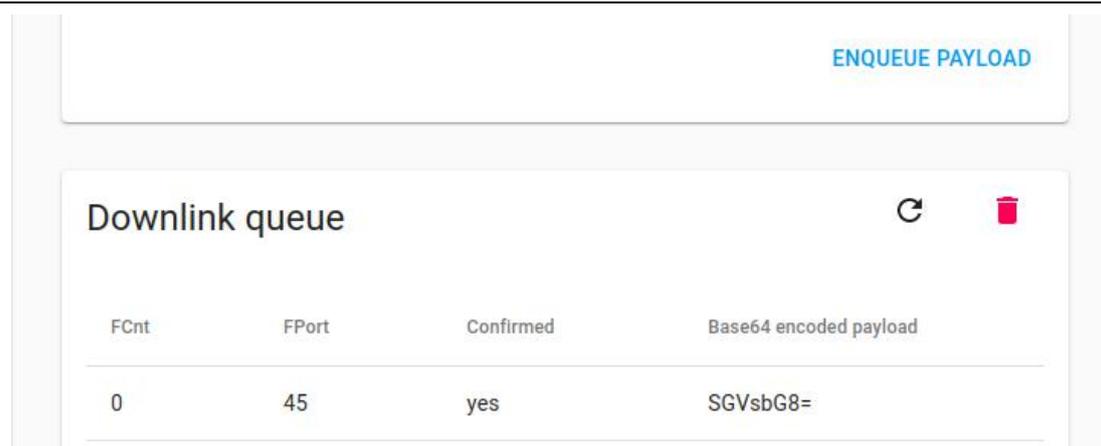
Confirmed downlink

BASE64 ENCODED JSON OBJECT

Base64 encoded string *

[ENQUEUE PAYLOAD](#)

- 3) 填写完成之后，点击“ENQUEUE PAYLOAD”，下行数据就会缓存到队列中



4) 节点再次成功发送一条数据到服务器之后，节点即可接收到刚刚服务器下发的数据。

“48656c6c6f”是“Hello”的十六进制形式。“48”是“H”的16进制ascii码。

```

[16:35:31.625]发->◇at+send=lor:1:48656c6c6f
□
[16:35:32.850]收←◆OK
at+recv=45,-33,7,5:48656c6c6f
  
```

3 OTAA CLASS C

3.1 创建 Otaa class C 的 Device-profile

首先我们创建 Otaa class c 的 Device-profile，该 Device-profile 只需要配置以下三页内容：

Device-profiles / dev-profile_otaa_class_c

GENERAL

JOIN (OTAA / ABP)

CLASS-B

CLASS-C

Device-profile name *

dev-profile_otaa_class_c

A name to identify the device-profile.

LoRaWAN MAC version *

1.0.2

The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision *

A

Revision of the Regional Parameters specification supported by the device.

Max EIRP *

0

Maximum EIRP supported by the device.

Geolocation buffer TTL (seconds)

0

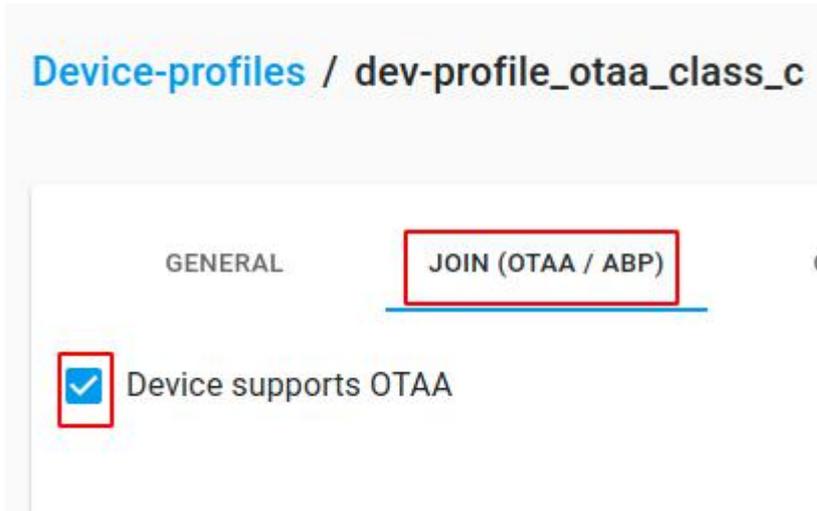
The time in seconds that historical uplinks will be stored in the geolocation buffer.

Geolocation minimum buffer size

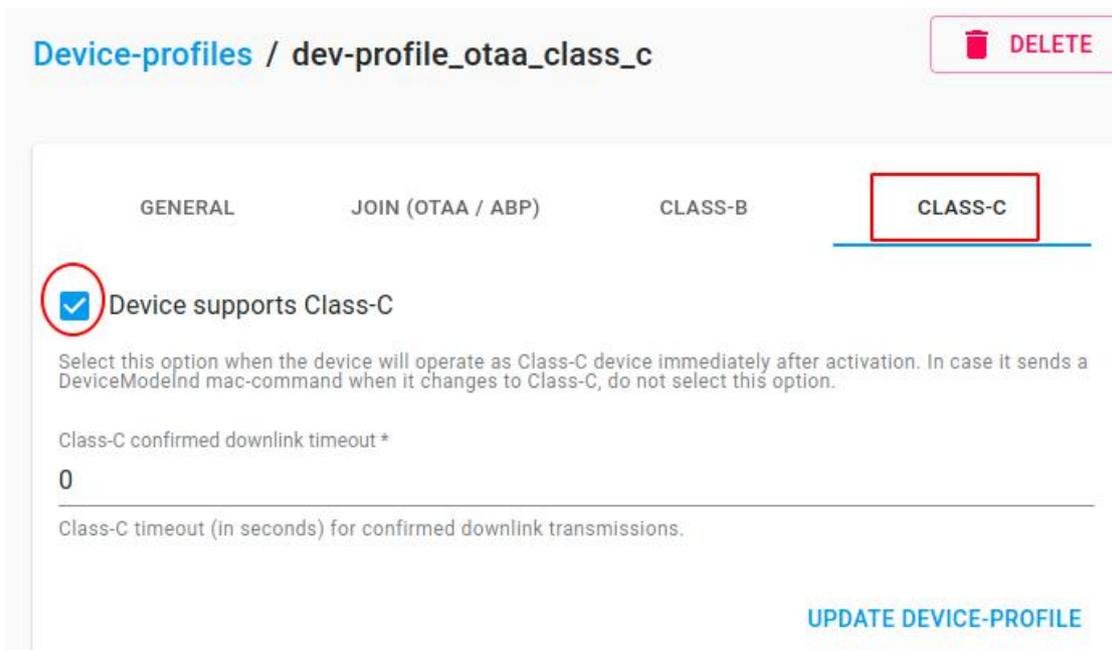
0

The minimum buffer size required before using geolocation (when enabled in the Service Profile). Usin

与



勾选 class c



3.2 创建 Otaa class C 的 Device

创建 Otaa class C 的 device 可参考上一章节的[创建 Otaa class A 的 Device](#) 内容，但是要注意，选择 device profile 的时候需要选择我们刚刚创建的 Otaa class C 的 device profile。

3.3 设置节点信息

使用上一步生成的 device EUI 和 Application key 设置我们节点的三元组信息。对于 ChirpStack 服务器，节点只需要设置 device EUI 和 Application key 即可，对于 app EUI 无需设置，但请不要将 app EUI 设置为全 0。

```
at+set_config=lora:dev_eui:xxxxxxxxxxxxxxxx
```

```
at+set_config=lora:app_eui:0000000000000555
```

```
at+set_config=lora:app_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:class:2
```

```
at+set_config=lora:join_mode:0
```

```
at+set_config=lora:work_mode:0
```

以上 at 指令设置了节点的三元组信息，以及将节点设置为 otaa class c 模式。

设置成功之后，可以使用 `at+get_config=lora:status` 检查是否设置成功。

3.4 服务器下发数据到节点

- 1) 节点需要先 join 成功，然后成功发送一条数据到服务器。

```

[15:23:57.041]发->◇at+join
□
[15:24:02.237]收<-◆OK Join Success
[15:24:39.240]发->◇at+send=lora:1:48656c6c6f
□
[15:24:40.352]收<-◆OK
at+recv=0,-96,7,0
  
```

- 2) 在 ChirpStack 页面中点开我们刚刚创建的 Device。Port 取值范围是 1-255, Confirmed downlink 是可选项，“SGVsbG8=” 是 “Hello” 的 base64 编码

Enqueue downlink payload

Port*
67

Please note that the fPort value must be > 0.

Confirmed downlink

BASE64 ENCODED JSON OBJECT

Base64 encoded string*

SGVsbG8=

ENQUEUE PAYLOAD

- 3) 填写完成之后，点击“ENQUEUE PAYLOAD”，下行数据就会缓存到队列中并立即发送给节点。

```

[15:54:00.210]发->◇at+join
□
[15:54:05.897]收←◆OK Join Success
[15:54:10.541]发->◇at+send=Lora:1:48656c6c6f
□
[15:54:11.649]收←◆OK
at+recv=0,-93,7,0
[15:54:25.848]收←◆at+recv=67,-94,8,5:48656c6c6f
  
```

4 ABP CLASS A

4.1 创建 ABP class A 的 Device-profile

首先我们创建 ABP class A 的 Device-profile，该 Device-profile 只需要配置下页内容：

Device-profiles / dev-profile_abp_class_a

GENERAL

JOIN (OTAA / ABP)

CLASS-B

Device-profile name *

dev-profile_abp_class_a

A name to identify the device-profile.

LoRaWAN MAC version *

1.0.2

The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision *

A

Revision of the Regional Parameters specification supported by the device.

Max EIRP *

0

Maximum EIRP supported by the device.

Geolocation buffer TTL (seconds)

0

The time in seconds that historical uplinks will be stored in the geolocation buffer.

Geolocation minimum buffer size

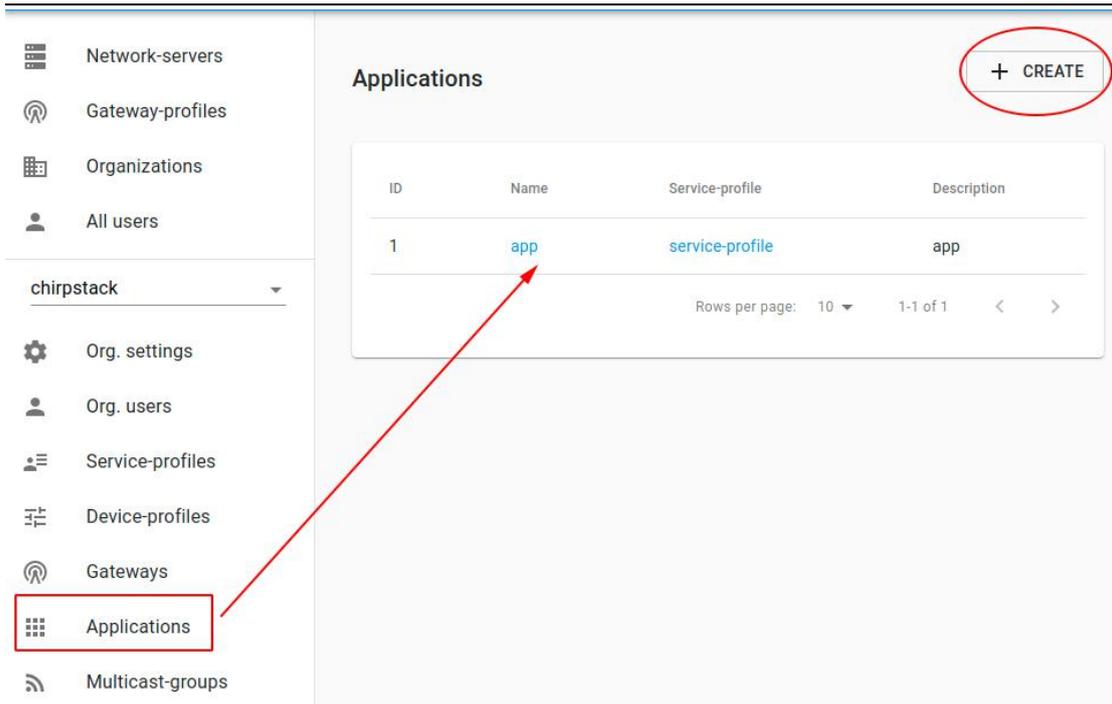
0

The minimum buffer size required before using geolocation (when enabled in the Servi

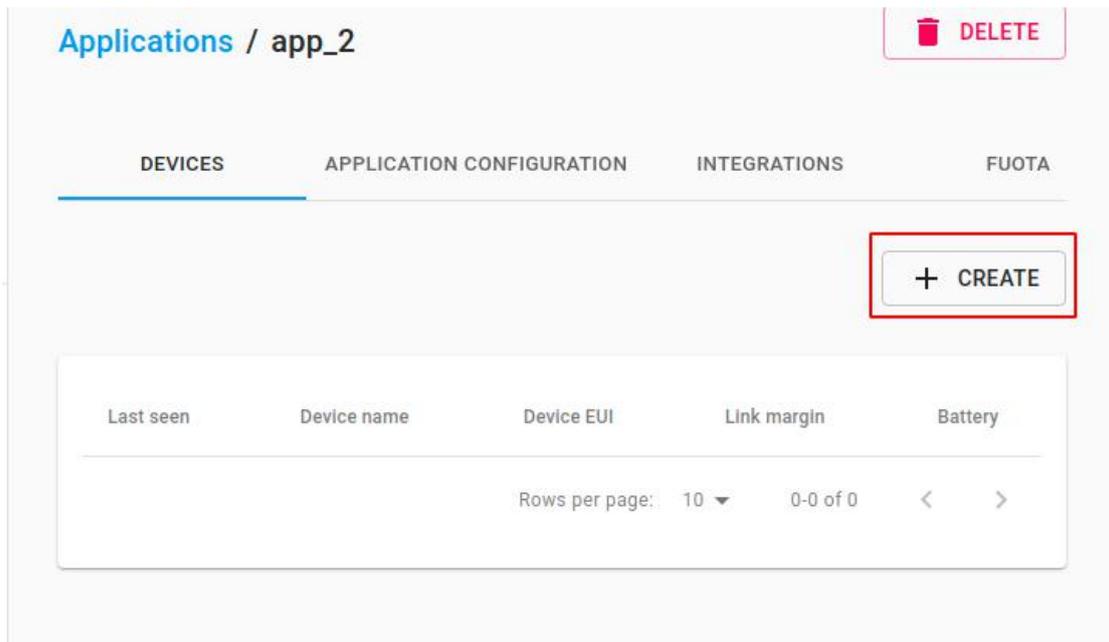
4.2 创建 ABP class A 的 Device

创建完成 Device-profile 之后，下面我们创建一个 abp class a 的节点。

- 1) 点击 Applications，打开一个应用。



2) 在打开的 application 中，点击右上角的 CREATE，创建一个节点



3) 如下图所示依次填入相应内容。Device EUI 我们可以自行填写，也可以点击按钮随机生成。Device-profile 中我们必须选择刚刚创建的 abp class a 的 device-profile。

Applications / app_2 / Devices / Create

GENERAL VARIABLES TAGS

Device name *
dev_abp_class_a

The name may only contain words, numbers and dashes.

Device description *
dev_abp_class_a

Device EUI *
6d d7 50 0f 43 0c e3 20 MSB ↻

Device-profile *
Device-profile

- device_profile_abp
- device_profile_otaa
- dev-profile_abp_class_a**
- dev-profile_abp_class_c

4) 请勾选 “Disable frame-counter validation”

Disable frame-counter validation

5) 填写完相应内容之后，点击右下角的 CREATE DEVICE，跳转到生成 ACTIVATION 页面。

点击按钮可生成随机的 Device address、Network session key、Application session key，
此处需要用户保存 Device address、Network session key、Application session key
以便后续使用：

< DETAILS
CONFIGURATION
KEYS (OTAA)
ACTIVATION
>

Device address *

00 51 5e 79 MSB ↻

Network session key (LoRaWAN 1.0) *

a6 46 78 20 0b 09 c9 de 78 64 49 6d 30 1f 86 b0 MSB ↻ 📄 🗑️

Application session key (LoRaWAN 1.0) *

bf c4 96 58 58 30 d0 bd fd 1c 5e 9b 4e bf e7 60 MSB ↻ 📄 🗑️

Uplink frame-counter *

0

Downlink frame-counter (network) *

0

(RE)ACTIVATE DEVICE

4.3 设置节点信息

使用上一步生成的 Device address、Network session key、Application session key 设置

我们节点的配置信息。

```
at+set_config=lora:dev_addr:xxxx
```

```
at+set_config=lora:nwks_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:apps_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:class:0
```

```
at+set_config=lora:join_mode:1
```

```
at+set_config=lora:work_mode:0
```

以上 at 指令设置了节点的信息，以及将节点设置为 abp class a 模式。

设置成功之后，可以使用 `at+get_config=lora:status` 检查是否设置成功。

4.4 服务器下发数据到节点

- 1) 节点需要先 join 成功，然后成功发送一条数据到服务器。

```

[15:23:57.041]发→◇at+join
□
[15:24:02.237]收←◆OK Join Success
[15:24:39.240]发→◇at+send=lora:1:48656c6c66f
□
[15:24:40.352]收←◆OK
at+recv=0,-96,7,0
  
```

- 2) 在 ChirpStack 页面中点开我们刚刚创建的 Device。Port 取值范围是 1-255, Confirmed downlink 是可选项，“SGVsbG8=” 是 “Hello” 的 base64 编码

Enqueue downlink payload

Port *

Please note that the fPort value must be > 0.

Confirmed downlink

BASE64 ENCODED JSON OBJECT

Base64 encoded string *

[ENQUEUE PAYLOAD](#)

- 3) 填写完成之后，点击“ENQUEUE PAYLOAD”，下行数据就会缓存到队列中。

ENQUEUE PAYLOAD

Downlink queue




FCnt	FPort	Confirmed	Base64 encoded payload
0	45	yes	SGVsbG8=

4) 节点成功发送一条数据到服务器之后，即可接收到服务器下发的数据。

```

[16:35:31.625]发->◇at+send=lora:1:48656c6c6f
□
[16:35:32.850]收←◆OK
at+recv=45,-33,7,5:48656c6c6f
  
```

5 ABP CLASS C

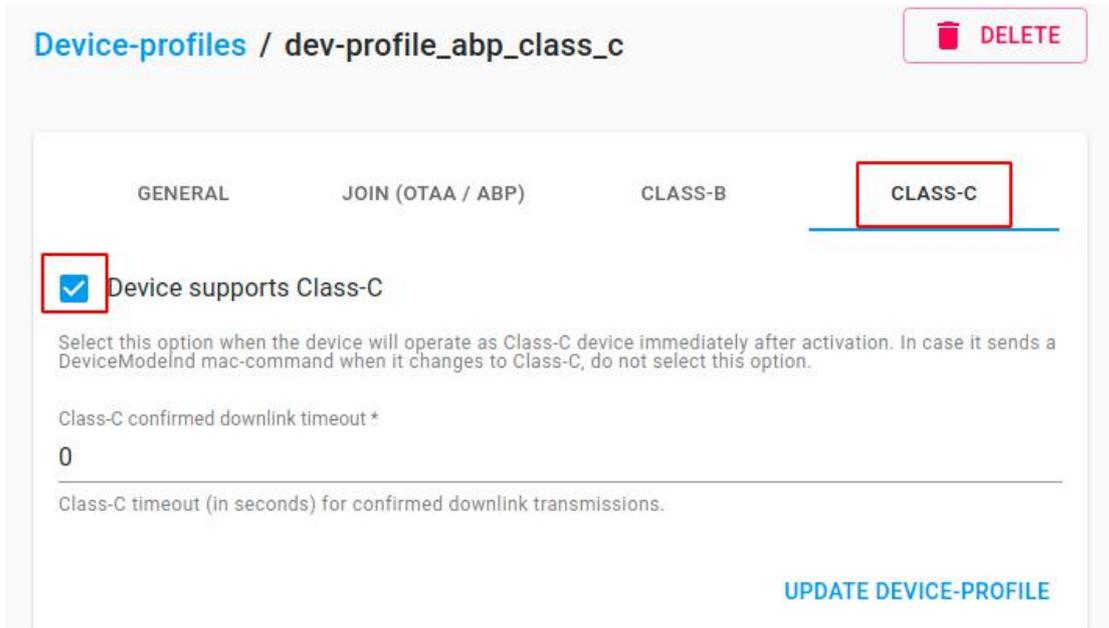
5.1 创建 ABP class C 的 Device-profile

首先我们创建 ABP class C 的 Device-profile，该 Device-profile 只需要配置以下两页内容：

Device-profiles / dev-profile_abp_class_c

 DELETE

GENERAL	JOIN (OTAA / ABP)	CLASS-B	CLASS-C
Device-profile name *			
dev-profile_abp_class_c			
A name to identify the device-profile.			
LoRaWAN MAC version *			
1.0.2			
The LoRaWAN MAC version supported by the device.			
LoRaWAN Regional Parameters revision *			
A			
Revision of the Regional Parameters specification supported by the device.			
Max EIRP *			
0			
Maximum EIRP supported by the device.			
Geolocation buffer TTL (seconds)			
0			
The time in seconds that historical uplinks will be stored in the geolocation buffer.			
Geolocation minimum buffer size			
0			
The minimum buffer size required before using geolocation (when enabled in the Service Profile). Using multiple uplinks for geolocation can increase the accuracy of the geolocation results.			



5.2 创建 ABP class C 的 Device

创建 ABP class C 的 device 可参考上一章节的[创建 ABP class A 的 Device](#) 内容，但是要注意，选择 device profile 的时候需要选择我们刚刚创建的 ABP class C 的 device profile。

5.3 设置节点信息

使用上一步生成的 Device address、Network session key、Application session key 设置我们节点的配置信息。

```
at+set_config=lora:dev_addr:xxxx
```

```
at+set_config=lora:nwks_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:apps_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
at+set_config=lora:class:2
```

```
at+set_config=lora:join_mode:1
```

```
at+set_config=lora:work_mode:0
```

以上 at 指令设置了节点的信息，以及将节点设置为 abp class c 模式。

设置成功之后，可以使用 `at+get_config=lora:status` 检查是否设置成功。

5.4 服务器下发数据到节点

- 1) 节点需要先 join 成功，然后成功发送一条数据到服务器。

```

[15:23:57.041]发->◇at+join
□
[15:24:02.237]收←◆OK Join Success
[15:24:39.240]发->◇at+send=lora:1:48656c6c6f
□
[15:24:40.352]收←◆OK
at+recv=0,-96,7,0
  
```

- 2) 在 ChirpStack 页面中点开我们刚刚创建的 Device。Port 取值范围是 1-255, Confirmed downlink 是可选项，“SGVsbG8=” 是 “Hello” 的 base64 编码

Enqueue downlink payload

Port *

34

Please note that the fPort value must be > 0.

Confirmed downlink

BASE64 ENCODED JSON OBJECT

Base64 encoded string *

SGVsbG8=

ENQUEUE PAYLOAD

Downlink queue ↻ 🗑

FCnt	FPort	Confirmed	Base64 encoded payload
------	-------	-----------	------------------------

- 3) 填写完成之后，点击“ENQUEUE PAYLOAD”，下行数据就会缓存到队列中并立即发送给节点。

```
[17:09:46.929]收←◆at+recv=34,-30,6,5:48656c6c6f
```

6 附录一：使用中可能遇到的问题

如果 ABP class c 模式下，节点无法接收到服务器下发的数据，请按以下步骤依次操作。abp class a 模式直接跳转第 3 步。

- 1) 节点输入 `at+get_config=lora:status` 查看输出

```
[17:45:14.395]发→◇at+get_config=lora:status
□
[17:45:14.451]收←◆OK Work Mode: LoRaWAN
Region: AS923
MulticastEnable: false
DutycycleEnable: false
Send_repeat_cnt: 1
Join_mode: ABP
DevAddr: 0135C218
AppsKey: 9A4631C87B6747765BFE6BA9A9B42F6E
NwksKey: E26C3790AF9B070B139B77B147749356
Class: C
Joined Network: true
IsConfirm: confirm
AdrEnable: true
EnableRepeaterSupport: false
RX2_CHANNEL_FREQUENCY: 923200000, RX2_CHANNEL_DR: 2
RX_WINDOW_DURATION: 3000ms
RECEIVE_DELAY_1: 1000ms
RECEIVE_DELAY_2: 2000ms
JOIN_ACCEPT_DELAY_1: 5000ms
JOIN_ACCEPT_DELAY_2: 6000ms
Current Datarate: 5
Primeval Datarate: 5
ChannelsTxPower: 0
UpLinkCounter: 2
DownLinkCounter: 3
```

- 2) 如果是 ABP class c 模式，请尝试修改 ABP class c 的 device profile 文件。根据节点 `at+get_config=lora:status` 输出的信息修改 device profile 文件。

Device-profiles / dev-profile_abp_class_c DELETE

GENERAL **JOIN (OTAA / ABP)** CLASS-B CLASS-C

Device supports OTAA

RX1 delay *
0

RX1 delay (valid values are 0 - 15).

RX1 data-rate offset *
0

Please refer the LoRaWAN Regional Parameters specification for valid values.

RX2 data-rate *
2

Please refer the LoRaWAN Regional Parameters specification for valid values.

RX2 channel frequency (Hz) *
923200000

Factory-preset frequencies (Hz) *
923200000

List of factory-preset frequencies (Hz), comma separated.

[UPDATE DEVICE-PROFILE](#)

- 3) 在 application 中打开对应的 device，将 Uplink frame-counter 和 Downlink frame-counter 设置为 0，然后点击(RE)ACTIVATE DEVICE 按钮

Applications / app / Devices / dev_abp_class_c DELETE

DETAILS CONFIGURATION KEYS (OTAA) **ACTIVATION** DEVICE DATA >

Device address *
01 35 c2 18 MSB ↻

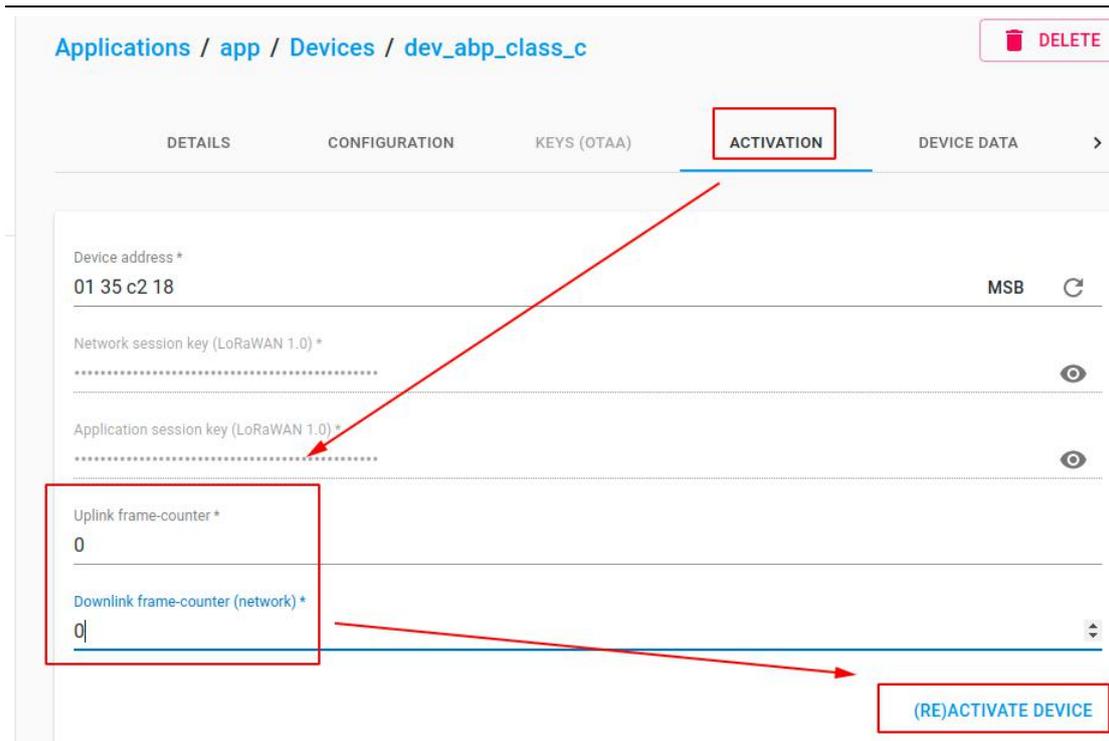
Network session key (LoRaWAN 1.0) *
..... 🔒

Application session key (LoRaWAN 1.0) *
..... 🔒

Uplink frame-counter *
0

Downlink frame-counter (network) *
0

(RE)ACTIVATE DEVICE



4) 重启节点并重新 join。重新执行服务器下发数据。