





# 3GPP

Item	Reference
CHF NF	3GPP TS 23.501 6.2.16
Nchf_ConvergedCharging	3GPP TS 32.291
	3GPP TS 32.291 6.1.3.2.1
	3GPP TS 32.291 6.1.3.2.2
	3GPP TS 32.291 6.1.3.2.3
ChargingDataRequest / Response	3GPP TS 32.291 6.1.6
5G PDU CDR	3GPP TS 32.290
SBI	3GPP TS 29.500
NRF NF	3GPP TS 29.510

## SBI

URI: /nchf-convergedcharging/v3

Method	Endpoint	Description
POST	/chargingdata	<p>Creates a new charging data record.</p> <p>chargingDataRef: Reference to the charging data record.</p> <p>CGRateS: Charging rate structure.</p>
POST	/chargingdata/{chargingDataRef}/update	<p>Updates the charging data record.</p> <p>chargingDataRef: Reference to the charging data record.</p>
POST	/chargingdata/{chargingDataRef}/release	<p>Releases the charging data record.</p> <p>chargingDataRef: Reference to the charging data record.</p> <p>CDR: Call Detail Record.</p> <p>CGRateS: Charging rate structure.</p>



# ChargingDataResponse — 0000

Field	Type	Format	Description
<code>invocationSequenceNumber</code>	integer	0 00 00	00000000000000000000 100 CHF-M10000000000000000
<code>invocationResult</code>	object	0 00 00	0000000000 {"resultCode": "SUCCESS"}0
<code>sessionId</code>	string	0 00 00	00000000 chargingDataRef (UUID)0
<code>multipleUnitInformation</code>	array	0 00 00	000000000000000000 resultCode grantedUnit (totalVolume, time) 0 ratingGroup (0000 100 CHF-L2)0

0000

00000000000000000000 `config/runtime.exs`00

```
config :omnichf,  
  sbi_scheme:      "http",  
  sbi_addr:        "127.0.0.14",  
  sbi_port:        7777,  
  nrf_uri:         "http://127.0.0.10:7777",  
  mcc:            "999",  
  mnc:            "70",  
  heartbeat_interval: 10_000,  
  cgrates_enabled: false,  
  cgrates_url:     "http://localhost:2080/jsonrpc",  
  cgrates_tenant:  "cgrates.org",  
  cgrates_timeout: 5000
```

□□□

□□	□□□	□□	
<code>sbi_scheme</code>	<code>"http"</code>	string	SBI
<code>sbi_addr</code>	<code>"127.0.0.14"</code>	string	SBI   □□
<code>sbi_port</code>	<code>7777</code>	integer	SBI   □□
<code>nrf_uri</code>	<code>"http://127.0.0.10:7777"</code>	string	□□ M URIG
<code>mcc</code>	<code>"999"</code>	string	□□□□ □□□□
<code>mnc</code>	<code>"70"</code>	string	□□□□ □□□□
<code>heartbeat_interval</code>	<code>10_000</code>	integer (ms)	NRF
<code>cgrates_enabled</code>	<code>false</code>	boolean	CGR <code>false</code> CGR 86,4 CGR <code>true</code>
<code>cgrates_url</code>	<code>"http://localhost:2080/jsonrpc"</code>	string	CGR RPC <code>cgra</code> <code>true</code>

名前	値	型	説明
cgrates_tenant	"cgrates.org"	string	CGRates Sess Tena CGR
cgrates_timeout	5000	integer (ms)	CGR HT min 3000

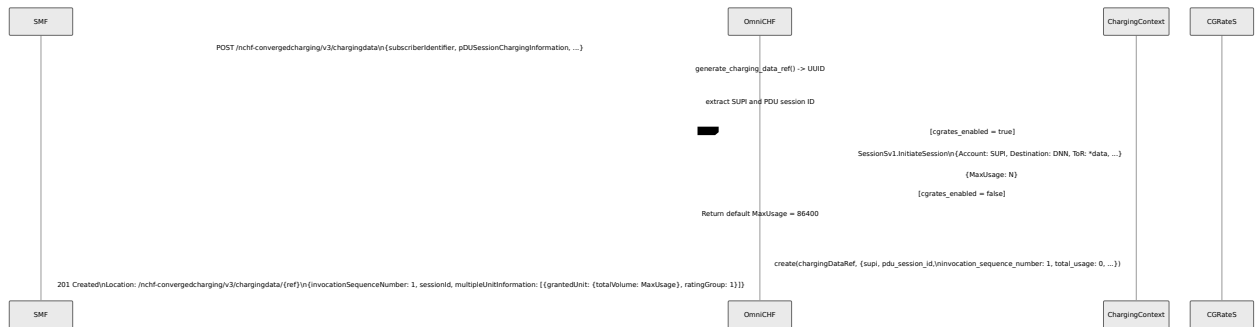
## CGRates 設定

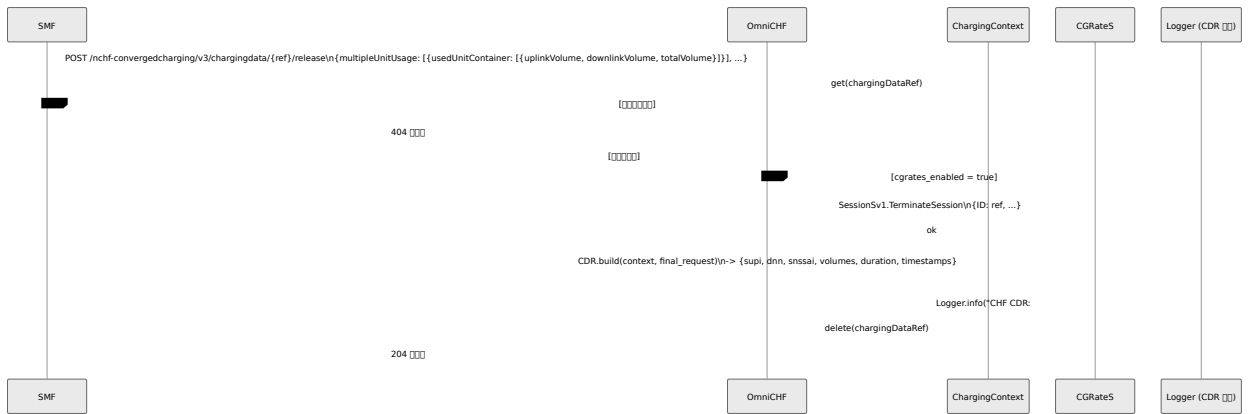
cgrates\_enabled false OmniCHF 86,400 CGRateS

CGRates Erlang :httpc HTTP CHF-M5 CGRateS HTTP

設定

設定





# CGRateS

OmniCHF 5G CGRateS SessionS

CGRateS		
Account	subscriberIdentifier (SUPI)	nfConsum
Subject	subscriberIdentifier (SUPI)	Account
Destination	pduSessionInformation.dnnId .dnn	
ToR	pduSessionInformation.pduType	PDU
RequestType	requestType	*p
Usage	usedUnitContainer.totalVolume + time	
OriginID	chargingDataRef	
OriginHost	"OmniCHF"	
SUPI	subscriberIdentifier	5G
DNN	pduSessionInformation.dnnId	5G
S-NSSAI_SST	pduSessionInformation.sNSSAI.sst	5G
S-NSSAI_SD	pduSessionInformation.sNSSAI.sd	5G
5QI	pduSessionInformation.qoSInformation.5qi	5G
RATType	pduSessionInformation.ratType	"NR"
PDUSESSIONID	pduSessionInformation.pduSessionID	5G
PDUSESSIONTYPE	pduSessionInformation.pduType	"IPV"

# CDR 表

CDR 表のスキーマ INFO 表のスキーマ CDR 表

フィールド名	データ型
record_type	enum "5G_PDU_SESSION"
supi	string
dnn	pduSessionInformation.dnnId   .dnn
snsai	{sst, sd}   pduSessionInformation.sNSSAI
qos_5qi	pduSessionInformation.qoSInformation.5qi
rat_type	pduSessionInformation.ratType
pdu_session_id	string
pdu_session_type	pduSessionInformation.pduType
volume_uplink	usedUnitContainer.uplinkVolume
volume_downlink	usedUnitContainer.downlinkVolume
volume_total	usedUnitContainer.totalVolume (uplink+downlink)
duration	usedUnitContainer.time (seconds)
start_time	created_at
end_time	string
charging_data_ref	UUID

---

□□□□

□□□□

□□	□□	□□	□□
<code>[:omnichf, :charging, :initial]</code>	count	supi	□□□□□□□□
<code>[:omnichf, :charging, :update]</code>	count	ref	□□□□□□□□
<code>[:omnichf, :charging, :release]</code>	count	ref	□□□□□□□□
<code>[:omnichf, :charging, :creates]</code>	count	result (success/failure)	□□□□□□
<code>[:omnichf, :charging, :updates]</code>	count	result	□□□□□□
<code>[:omnichf, :charging, :releases]</code>	count	result	□□□□□□
<code>[:omnichf, :cgrates, :request]</code>	count, duration_ms	operation, result	□□ CGRateS JSON-RPC □ □
<code>[:omnichf, :cgrates,</code>	status (1/0)	—	CGRateS □□ □□

名前	タイプ	ラベル	説明
<code>:health]</code>			
<code>[:omnichf, :sessions, :active]</code>	count	—	セッション数
<code>[:omni5g, :nrf, :registration]</code>	status (1/0)	nf_type	NRF ステータス

## Prometheus

メトリクス

メトリクス名	タイプ	ラベル	説明
<code>omni_chf.charging.initial.count</code>	counter	supi	初期充電回数
<code>omni_chf.charging.update.count</code>	counter	ref	更新回数
<code>omni_chf.charging.release.count</code>	counter	ref	リリース回数
<code>omni_chf.charging.create.total</code>	counter	result	作成結果
<code>omni_chf.charging.update.total</code>	counter	result	更新結果
<code>omni_chf.charging.release.total</code>	counter	result	リリース結果
<code>omni_chf.sessions.active.count</code>	gauge	--	アクティブセッション数

## CGRateS

名前	タイプ	メソッド	説明
<code>omni_chf.cgrates.calls.count</code>	counter	<code>method</code> , <code>result</code>	CGF JSON RPC
<code>omni_chf.cgrates.latency.milliseconds</code>	gauge	--	CGF JSON
<code>omni_chf.cgrates.health</code>	gauge	--	CGF JSON (1=健康 0=不健康)
<code>omni_chf.cgrates.requests.total</code>	counter	<code>operation</code> , <code>result</code>	CGF JSON RPC
<code>omni_chf.cgrates.request.duration_ms</code>	distribution	<code>operation</code>	CGF JSON 10, 50, 250 500 1000 2500

## NRF

名前	タイプ	メソッド	説明
<code>omni_chf.nrf.registration.status</code>	gauge	<code>nf_type</code>	NRF 登録状態 (1=登録済 0=未登録)

## BEAM VM

名前	タイプ	説明
<code>beam.memory.total</code>	gauge	BEAM の総メモリ使用量
<code>beam.memory.processes</code>	gauge	Erlang のプロセス数
<code>beam.memory.processes_used</code>	gauge	プロセスに使用されているメモリ
<code>beam.memory.system</code>	gauge	システムメモリ使用量
<code>beam.memory.atom</code>	gauge	アトム数
<code>beam.memory.atom_used</code>	gauge	アトムに使用されているメモリ
<code>beam.memory.binary</code>	gauge	バイナリ数
<code>beam.memory.code</code>	gauge	コード数
<code>beam.memory.ets</code>	gauge	ETS のメモリ使用量
<code>beam.processes.count</code>	gauge	Erlang のプロセス数
<code>beam.ports.count</code>	gauge	Erlang のポート数
<code>beam.atom.count</code>	gauge	アトム数
<code>beam.vm.uptime</code>	gauge	VM の稼働時間

□□□□

□□	□□	□□
info	CHF Create: ref=<UUID> supi=<SUPI> pdu_session=<N>	□□□□□□
info	CHF Update: ref=<UUID>	□□□□□□
info	CHF Release: ref=<UUID>	□□□□□□
info	CHF CDR: %{...}	□□□□□□ CDR
info	Initiating CGRateS session for <ref>, account: <SUPI>	□□ CGRateS InitiateSession
info	CGRateS authorized <N> units for session <ref>	□□□□
info	CGRateS session <ref> terminated successfully	CGRateS TerminateSession OK
warning	CGRateS integration disabled, returning default authorization	□□□□□□
warning	CHF Update: unknown ref=<UUID>	□□□□□□□□□□
warning	CHF Release: unknown ref=<UUID>	□□□   □□□□□□
error	CHF Create failed: <reason>	□□□□□□
error	CHF Update failed: <reason>	□□□□□□
error	CGRateS InitiateSession failed for <ref>: <reason>	□□□ CGRateS □□

消息	消息内容	消息类型
error	CGRateS HTTP error <status>: <body>	CGRateS 消息 200
error	CGRateS HTTP request failed: <reason>	CGRateS 消息

---





2. `"CHF CDR:"` `chargingID` `recordingEntity` `CHF-L4` `ratingGroup` `1` `CHF-L2`
3. CDR `chargingID` `recordingEntity` `CHF-L4` `ratingGroup` `1` `CHF-L2`

## CGRateS

`CGRateS` `:httpc` `CHF-M5`

1. `omni_chf.cgrates.request.duration_ms` p99
2. OmniCHF
3. `cgrates_timeout` CGRateS Elixir

`omni_chf.sessions.active.count`

1. 404 SMF 404 SMF OmniCHF
2. OmniCHF SMF 404

## NRF

`omni_chf.nrf.registration.status` 0

1. `nrf_uri` NRF OmniCHF `sbi_addr`
2. `mcc` `mnc` NRF PLMN
3. NRF