← نظرة عامة

مرجع تكوين OmniCHF

يتم قراءة تكوين المشغل من مفتاح بيئة التطبيق `:omnichf`، الذي يتم تعبئته عند بدء التشغيل. يمكن تجاوز كل مفتاح بواسطة متغير بيئة نظام التشغيل بواسطة `config/runtime.exs`. المثال أدناه يظهر التكوين الكامل في وقت التشغيل مع القيم المطبقة عندما يكون المتغير البيئي المقابل غير مضبوط.

```
config :omnichf,
  # SBI (واجهة قائمة على الخدمة) المستمع
  sbi_scheme: "http",
  sbi_addr:   "127.0.0.14",
  sbi_port:   7777,

  # تسجيل NRF / نبض القلب
  nrf_uri:    "http://127.0.0.1:7777",

  # هوية PLMN المقدمة
  mcc: "999",
  mnc: "70",

  # المراقبة
  prometheus_metrics_port: 9568,

  # تردد نبض القلب NRF
  heartbeat_interval: 10_000,

  # رصيد / تقييم محرك تكامل CGRateS
  cgrates_enabled: true,
  cgrates_url:     "http://localhost:2080/jsonrpc",
  cgrates_tenant:  "cgrates.org",
  cgrates_timeout: 5000

# تسجيل JSON منظم
config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

المعلمات الأساسية

المعلمة	النوع	مطلوب	الافتراضي
`sbi_scheme`	سلسلة	لا	`"http"`
`sbi_addr`	سلسلة	لا	`"127.0.0.14"`
`sbi_port`	عدد صحيح	لا	`7777`
`nrf_uri`	سلسلة	لا	`"http://127.0.0.1:7777"`
`mcc`	سلسلة	لا	`"999"`

المعلمة	النوع	مطلوب	الافتراضي
`mnc`	سلسلة	لا	`"70"`
`prometheus_metrics_port`	عدد صحيح	لا	`9568`
`heartbeat_interval`	عدد صحيح (مللي ثانية)	لا	`10000`

معلمات CGRateS

تتحكم هذه المفاتيح في التكامل مع محرك تقييم ورصيد CGRateS الخارجي. انظر تكامل للحصول على تفاصيل السلوك CGRateS.

المعلمة	النوع	مطلوب	الافتراضي
`cgrates_enabled`	منطقي	لا	`true` (`"true"`)
`cgrates_url`	سلسلة	لا	`"http://localhost:2080/jsonrpc"`
`cgrates_tenant`	سلسلة	لا	`"cgrates.org"`

المعلمة	النوع	مطلوب	الافتراضي
`cgrates_timeout`	عدد صحيح (مللي ثانية)	لا	`5000`

> **الافتراضي عند التعطيل في وقت البناء:** الافتراضي المترجم (المستخدم إذا لم يتم تعبئة بيئة التطبيق بواسطة `runtime.exs`) هو `cgrates_enabled: false` في نشر. يفرض ملف التعريف `true` بتعيينه، وهناك الافتراضي هو `runtime.exs` عادي، يقوم الاختباري `cgrates_enabled: false`.

المفاتيح المتقدمة / الاخ��يارية

هذه المفاتيح ليست مملوءة بواسطة `runtime.exs` الافتراضي ولكن يتم قراءتها من بيئة إذا كانت موجودة. إنها تضبط دقة مجموعة التقييم، الشحن غير المتصل، وتوليد معرف `:omnichf` الشحن. ليس لديها متغير بيئة مخصص ويتم تعيينها مباشرة في كتلة تكوين `:omnichf`.

المعلمة	النوع	مطلوب	الافتراضي
`rating_groups`	خريطة	لا	`%{}`
`default_rating_group`	عدد صحيح	لا	`1`
`offline_charging_enabled`	منطقي	لا	`false`
`cdr_output_dir`	سلسلة	لا	`"/var/log/omnichf/cdr"`
`node_id`	عدد صحيح	لا	`1`

مجموعات التقييم

تقوم `rating_groups` بربط **DNN** (القيمة، (مفتاح الخريطة، سلسلة) بـ **مجموعة التقييم**. تظهر مجموعة التقييم المحلولة في حقل DNN. عدد صحيح) المطبقة على جلسات ذلك المعادة للمستهلك وفي حقل CDR `multipleUnitInformation` من `ratingGroup` الدقة هي: `rating_group`.

1. إذا كان DNN في الجلسة مفتاحًا `rating_groups`، استخدم تلك القيمة.
2. خلاف ذلك استخدم `default_rating_group`.

```
config :omnichf,
  rating_groups: %{
    "internet" => 10,
    "ims"      => 20,
    "iot"      => 30
  },
  default_rating_group: 1
```

يتم المطابقة على سلسلة DNN الدقيقة. DNN بدون إدخال (ودون `default_rating_group`) غير الصحيحة (ليست خريطة) يتراجع إلى مجموعة التقييم `rating_groups` `1`. يتم تجاهل قيمة ويتم استخدام الافتراضي، لذا فإن تكوينًا خاطئًا لا يمنع الشحن أبدًا.

التسجيل

يتم تكوين تسجيل JSON المنظم عبر `:logger` `:default_formatter`.

المعلمة	النوع	الافتراضي	الوصف
`format`	مجموعة	`{OmniLogger.JsonFormatter, :format}`	مُنسق خط السجل. يصدر JSON منظم، مناسب للاستهلاك بواسطة شاحن السجلات.
`metadata`	ذرة / قائمة	`:all`	بيانات التعريف الخاصة بالسجل المضمنة في كل سطر. `:all` يصدر جميع بيانات التعريف المرفقة (SUPI، DNN، معرف جلسة PDU، معرف الشحن، الإجراء، إلخ).

واجهات الإدارة

يعرض OmniCHF واجهات إدارة موجهة للمشغل بجانب SBI. هذه المنافذ ذات صلة بالمراقبة ولكنها ليست جزءًا من خدمة OAM وNchf:

الواجهة	المنفذ	المخطط	الغرض
SBI (Nchf)	`sbi_port` (الافتراضي `7777`)	HTTP	حركة شحن 5G بين NF.
مقاييس Prometheus	`prometheus_metrics_port` (الافتراضي `9568`)	HTTP	نقطة نهاية سحب المقاييس.
واجهة برمجة التطبيقات الإدارية	`8443`	HTTPS	قراءات الحالة، جلسات / إحصائيات / قراءات صحة CGRateS، بحث وإجراءات CDR، OAM (تكوين، تفريغ مستوى CDR، السجل، إجهاض الجلسة، إعادة تسجيل NRF).
لوحة التحكم	`7443`	HTTPS	واجهة ويب للموارد، التكوين، الترخيص، والسجلات.

نقاط نهاية واجهة برمجة التطبيقات الإدارية / OAM

تقوم واجهة برمجة التطبيقات الإدارية على المنفذ `8443` (HTTPS) بكشف الأسطح التالية للمشغل:

الطريقة	المسار	الغرض
`GET`	`/api/status/api`	قراءة حالة API / الخدمة.
`GET`	`/api/status/license`	قراءة حالة الترخيص.
`GET`	`/api/status/nrf`	قراءة حالة تسجيل NRF.
`GET`	`/api/status/nf`	قراءة حالة NF (العقدة).
`GET`	`/api/sessions`	قائمة بجلسات الشحن النشطة.
`GET`	`/api/sessions/{id}`	فحص جلسة شحن واحدة بواسطة `charging_data_ref`.
`GET`	`/api/statistics`	قراءة إحصائيات الشحن (عدد الجلسات النشطة، إجمالي حجم الشحن، قابلية الوصول إلى CGRateS).
`GET`	`/api/health/cgrates`	قراءة صحة الاتصال بـ CGRateS.
`GET`	`/api/oam/config`	قراءة تكوين وقت التشغيل.
`PATCH`	`/api/oam/config/{id}`	تحديث مفتاح تكوين وقت التشغيل.
`POST`	`/api/oam/charging_session`	إجهاض جلسة شحن بواسطة `charging_data_ref`.
`GET`	`/api/oam/cdr`	بحث CDR المكتملة. معلمات الاستعلام: `supi`، `dnn`، `date_from`، `date_to` (جميعها اختيارية).
`POST`	`/api/oam/cdr/flush`	فرض تفريغ الاستخدام المتراكم كـ CDR لجلسة واحدة (`{"charging_data_ref": "<ref>"}`) أو جميع الجلسات المتعقبة (`{}` / `"all"`).

الطريقة	المسار	الغرض
`POST`	`/api/oam/log_level`	تغيير مستوى السجل في وقت التشغيل.
`POST`	`/api/oam/nrf/reregister`	بدء إعادة تسجيل NRF.

> **رؤية تخزين الجلسة:** تقدم الأسطح الإدارية الموجهة للجلسة (`/api/sessions`، `/api/statistics`، `/api/oam/charging_session`، `/api/oam/cdr/flush`) كل الحي بكل جلسة في عملية **عامل** لكل جلسة، ويتم SBI جلسة حية. يحتف❓❓ مسار شحن عكس كل جلسة عامل إلى **تخزين السياق** في الذاكرة عند الإنشاء والتحديث ويتم إزالتها عند التوقف. تستفسر الأسطح الإدارية عن سجل العمال أولاً وتعود إلى السياق، لذا فإن الجلسة الحية دائمًا مرئية وقابلة للتنفيذ. انظر تخزين الجلسة ورؤية الإدارة لمعرفة ما يعنيه غير المتصلة وهو CDR يقرأ ملفات (`/api/oam/cdr`) CDR ذلك من الناحية التشغيلية. بحث مستقل عن كلا التخزينين.

← نظرة عامة

مرجع مقاييس OmniCHF

تُعرض المقاييس على `prometheus_metrics_port` (`9568` افتراضي) عند `/metrics`.

مقاييس الشحن

المقياس: `omni_chf_charging_initial_count`
النوع: عداد
الوصف: عدد الطلبات المستلمة لجلسة الشحن INITIAL (إنشاء).
التسميات: `supi`

المقياس: `omni_chf_charging_update_count`
النوع: عداد
الوصف: عدد الطلبات المستلمة لتحديث جلسة الشحن.
التسميات: `ref`

المقياس: `omni_chf_charging_release_count`
النوع: عداد
الوصف: عدد الطلبات المستلمة لإصدار جلسة الشحن.
التسميات: `ref`

المقياس: `omni_chf_charging_creates_total`
النوع: عداد
الوصف: نتائج عملية الإنشاء.
التسميات: `result` (`success` / `failure`)

المقياس: `omni_chf_charging_updates_total`
النوع: عداد
الوصف: نتائج عملية التحديث.
التسميات: `result` (`success` / `failure`)

المقياس: `omni_chf_charging_releases_total`
النوع: عداد
الوصف: نتائج عملية الإصدار. تم الإصدار عبر SBI مع إنهاء CGRateS ناجح (`success`)، تم الإصدار عبر SBI حيث فشلت `TerminateSession` لـ CGRateS ولكن تم إنهاء CDR

(`failure`، `500` كـ OAM CDR-flush (`oam_flush`).
التسميات: `result` (`success` / `failure` / `oam_flush`)

المقياس: `omni_chf_sessions_active_count`
النوع: مقياس
الوصف: عدد جلسات الشحن النشطة المحتفظ بها حاليًا في الذاكرة. يتم إصداره في كل إنشاء وحذف لمتجر السياق، والذي يتضمن الآن كل جلسة عامل حية مكررة، لذا يتتبع المقياس العدد الحقيقي للجلسات النشطة.
التسميات: لا شيء

```
# معدل إنشاء جلسة الشحن
rate(omni_chf_charging_creates_total{result="success"}[5m])

# نسبة فشل الإنشاء
sum(rate(omni_chf_charging_creates_total{result="failure"}[5m]))
  / sum(rate(omni_chf_charging_creates_total[5m]))

# الجلسات النشطة حاليًا
omni_chf_sessions_active_count
```

مقاييس CGRateS

المقياس: `omni_chf_cgrates_requests_total`
النوع: عداد
الوصف: طلبات CGRateS JSON-RPC.
التسميات: `operation` (طريقة SessionS)، `result` (`success` / `failure`)

المقياس: `omni_chf_cgrates_request_duration_ms`
النوع: هيستوجرام
الوصف: مدة طلب CGRateS JSON-RPC بالمللي ثانية.
التسميات: `operation`
الدلاء: 5, 10, 25, 50, 100, 250, 500, 1000, 2500

المقياس: `omni_chf_cgrates_health`
النوع: مقياس
الوصف: صحة الاتصال بـ CGRateS (`1` = قابل للوصول، `0` = غير متصل/معطل).
التسميات: لا شيء

```
# زمن الاستجابة p99 لـ CGRateS حسب العملية
histogram_quantile(0.99,
  sum by (le, operation)
(rate(omni_chf_cgrates_request_duration_ms_bucket[5m])))

# معدل الأخطاء لـ CGRateS
sum(rate(omni_chf_cgrates_requests_total{result="failure"}[5m]))

# قابلية الوصول لـ CGRateS (تنبيه عند 0)
omni_chf_cgrates_health
```

مقاييس NRF

المقياس: `omni_chf_nrf_registration_status`
النوع: مقياس
الوصف: حالة تسجيل NRF (`1` = مسجل، `0` = غير مسجل).
التسميات: `nf_type`

```
omni_chf_nrf_registration_status
```

مقاييس BEAM VM

يتم تصدير مقاييس صحة وقت التشغيل القياسية أيضًا.

المقياس	النوع	الوصف
`beam_memory_total`	مقياس	إجمالي ذاكرة BEAM (بايت)
`beam_memory_processes` / `beam_memory_processes_used`	مقياس	ذاكرة العمليات المخصصة / المستخدمة
`beam_memory_system`	مقياس	ذاكرة النظام (غير العمليات)
`beam_memory_atom` / `beam_memory_atom_used`	مقياس	ذاكرة الذرات المخصصة / المستخدمة
`beam_memory_binary`	مقياس	ذاكرة ثنائية
`beam_memory_code`	مقياس	ذاكرة الشيفرة
`beam_memory_ets`	مقياس	ذاكرة جدول ETS
`beam_processes_count`	مقياس	عدد عمليات Erlang
`beam_ports_count`	مقياس	عدد منافذ Erlang
`beam_atom_count`	مقياس	عدد الذرات
`beam_vm_uptime`	مقياس	وقت تشغيل VM (ثواني)

← نظرة عامة

OmniCHF دليل عمليات

جدول المحتويات

1. نظرة عامة
2. دورة حياة جلسة الشحن
3. 3GPP مرجع دور والمواصفات
4. SBI نقاط نهاية
5. CGRateS تكامل
6. الإجراءات الرئيسية

نظرة عامة

تنفذ OmniCHF وظيفة الشحن **(CHF)** من الجيل الخامس. يوفر CHF شحنًا متقاربًا عبر الإنترنت وغير متصل لجلسات PDU من خلال خدمة **Nchf_ConvergedCharging** من الجيل الخامس (3GPP TS 32.290 / TS 32.291 / TS 32.255).

ينشئ مستهلك الشحن، عادةً **SMF**، جلسة شحن لكل جلسة PDU، ويبلغ عن الاستخدام على مدار عمر تلك الجلسة، وأخيرًا يحررها. تقوم OmniCHF بترجمة كل طلب شحن من الجيل الخامس إلى استدعاء JSON-RPC لجلسة **CGRateS** للحصول على تفويض الائتمان في الوقت الحقيقي (الشحن عبر الإنترنت / إدارة الحصص) وتولد **سجل بيانات الشحن (CDR)** عند تحرير الجلسة (الشحن غير المتصل).

تكون كل جلسة شحن مملوكة لـ **ChargingWorker** GenServer مخصص تحت `DynamicSupervisor` (بنية عملية لكل جلسة). يتم الاحتفاظ بجميع الحالة لجلسة واحدة (SUPI، معرف CGRateS، الحجم/المدة المتراكمة، مجموعة التصنيف، حالة جلسة S-NSSAI، DNN) في عملية العامل تلك. لا تؤثر انهيارات جلسة واحدة على الأخرى. يتم الاحتفاظ بحالة الشحن (الجلسة في الذاكرة ويتم إعادة بنائها عندما ينشئ المستهلكون جلسات شحن جديدة بعد إعادة التشغيل.

المسؤوليات الأساسية:

- خدمة `nchf-convergedcharging/v3` على SBI.
- تخصيص `chargingDataRef` لكل جلسة وإدارة دورة حياتها INITIAL → UPDATE → TERMINATE.
- منح وحدات الخدمة (GSU) استجابة لوحدات الخدمة المطلوبة (RSU)، وإرجاع الحجم/الوقت الممنوحين لإدارة الحصص إلى المستهلك **المحفزات**.
- الربط مع CGRateS (الشحن عبر الإنترنت) لإدارة التصنيف والتوازن.
- إصدار CDRs اليومية غير المتصلة إلى ملفات الشحن، اختياريًا، إلى سجل التطبيق.
- التسجيل مع NRF والحفاظ على نبض القلب.

دورة حياة جلسة الشحن

تتوافق كل جلسة PDU مع جلسة شحن واحدة، يتم تتبعها بواسطة `chargingDataRef` (UUID).

الحالة	الزناد	الإجراء
تم الإنشاء	`POST /chargingdata`	تم إنشاء ChargingWorker، تم استدعاء CGRateS `InitiateSession`، تم منح الحصة الأولية
تم التحديث	`POST /chargingdata/{ref}/update`	تم استدعاء CGRateS `UpdateSession`، تم تجميع الاستخدام، تم زيادة رقم التسلسل، تم منح حصة جديدة
تم التحرير	`POST /chargingdata/{ref}/release`	تم استدعاء CGRateS `TerminateSession`، تم بناء CDR، تم إيقاف العامل

تخزين الجلسة ورؤية الإدارة

تحافظ OmniCHF على حالة الجلسة النشطة في **مخزنين** في الذاكرة يتم الاحت��اظ بهما في تناغم، بحيث ترى واجهات الإدارة وتستطيع التصرف بناءً على كل جلسة حية:

المخزن	الدعم	يحتفظ بـ	الدور
سجل العامل	`ChargingWorker` GenServer لكل جلسة تحت `DynamicSupervisor`، مفاتيح في `Registry`	الحالة الكاملة لكل جلسة (الحجم/المدة المتراكمة، رقم التسلسل، حالة CGRateS)	مصدر الحقيقة لجلسة حية. الإنشاء يولد عاملاً؛ التحديث والتحرير يحلان العامل أولاً
مخزن السياق	`Agent` واحدة في الذاكرة خريطة مفاتيحها `chargingDataRef`	مرآة لنفس سمات الجلسة	نموذج قراءة للرؤية وواجهات OAM

الإنشاء دائمًا يولد عاملاً، و**التحديث** و**التحرير** يحلان ذلك العامل أولاً، والرجوع إلى SBI، على مخزن السياق فقط إذا لم يتم العثور على عامل (على سبيل المثال بعد انقطاع مشرف). يتم عكس كل جلسة عامل حية إلى مخزن السياق عند الإنشاء وعند كل تحديث، ويتم إزالتها منه عندما يتوقف العامل (ا��إفراج العادي، الإلغاء، أو الانهيار). تستفسر واجهات الإدارة عن سجل العامل أولاً وتعود إلى السياق، لذا يقدم كلا المخزنين عرضًا متسقًا.

عمليًا يعني هذا:

- الجلسة التي تم إنشاؤها بشكل طبيعي مرئية بالكامل من خلال واجهات الإدارة: تظهر في `GET /api/sessions` وفي إجماليات الإحصائيات، ويمكن إلغاؤها (`POST /api/oam/charging_session`) وإفراغ CDR (`POST /api/oam/cdr/flush`) من خلال نقاط نهاية OAM أثناء وجودها.
- تعكس مقياس الجلسة النشطة والإحصائيات وقوائم الجلسات مجموعة الجلسات الحية الحقيقية، وليس مجرد مجموعة احتياطية.
- يتم الاحتفاظ بكلا المخزنين في الذاكرة ويتم إعادة ملئهما عندما ينشئ المستهلكون جلسات جديدة بعد إعادة التشغيل. نظرًا لأن الجلسات محتفظ بها في سجل محلي، فإن `chargingDataRef` مملوك من قبل المثيل الذي أنشأه.

تؤكد مقياس `omni_chf_sessions_active_count` والعدادات `omni_chf_charging_creates_total` / `omni_chf_charging_releases_total` عدد الجلسات الحية المبلغ عنها بواسطة `GET /api/sessions`.

مرجع دور 3 GPP والمواصفات

المواصفة	الصلة
TS 23.501	مفهوم الشحن المتقارب، CHF هندسة نظام الجيل الخامس: دور
TS 32.240	هندسة الشحن والمبادئ
TS 32.290	هندسة الشحن في نظام الجيل الخامس وتدفقات الرسائل
TS 32.291	شحن نظام الجيل الخامس: واجهة برمجة تطبيقات خدمة **Nchf_ConvergedCharging** (HTTP/2 SBI)، نموذج بيانات ChargingDataRequest/Response
TS 32.255	شحن مجال الاتصال بالبيانات في الجيل الخامس (شحن جلسة PDU)
TS 32.251	تنسيق معرف الشحن (`chargingId`)، من TS 32.251 الفقرة 5.2.1.6
TS 32.298	أوصاف ومعالجة معلمات CDR
TS 29.500 / 29.501	إطار عمل مشترك API وتصميم SBI للجيل الخامس
TS 29.510	تسجيل NF ونبض القلب مع NRF

مراجع إجراء Nchf_ConvergedCharging (TS 32.291):

الإجراء	الفقرة
إنشاء بيانات الشحن (INITIAL)	6.1.3.2.1
تحديث بيانات الشحن (UPDATE)	6.1.3.2.2
تحرير بيانات الشحن (TERMINATE)	6.1.3.2.3
إشعار الشحن (REAUTHORIZATION / ABORT_CHARGING)	6.1.6.2

نقاط نهاية SBI

تُخدم جميع نقاط نهاية Nchf تحت عنوان URL الأساسي `{sbi_scheme}://{sbi_addr}:{sbi_port}` والمسار الأساسي `/nchf-convergedcharging/v3`.

الطريقة	المسار	الوصف
`POST`	`/chargingdata`	إنشاء جلسة شحن (**INITIAL**). يخصص `chargingDataRef`، يبدأ جلسة CGRateS، يمنح حصة أولية.
`POST`	`/chargingdata/{chargingDataRef}/update`	تحديث جلسة شحن (**UPDATE**). يبلغ عن الاستخدام ويطلب حصة إضافية.
`POST`	`/chargingdata/{chargingDataRef}/release`	تحرير جلسة شحن (**TERMINATE**). يبلغ عن الاستخدام النهائي، يولد CDR، ينهي جلسة CGRateS.
`POST`	`/notify`	إشعار شحن استدعاء. يأمر OmniCHF بإرسال إشعار `REAUTHORIZATION` أو `ABORT_CHARGING` إلى `notifyUri` المخزنة للمستهلك.

عندما ("العنوان "فشل إنهاء الشحن، `SYSTEM_FAILURE` السبب) عند التحرير، يتم إرجاع `500` وكتابته، ولكن يتم عرض CDR لا يزال يتم إنهاء. `TerminateSession` تفشل جلسة CGRateS (مفتوحة CGRateS جلسة) الفشل حتى لا يتم إخفاء عدم التناسق في الخلفية الفوترة. انظر إصدار يعود 500.

التحقق من الطلب

يتم التحقق من كل جسم طلب للحقائق المعلوماتية الإلزامية التالية قبل المعالجة. يؤدي IE المفقود إلى إرجاع `400 Bad Request` مع السبب `MANDATORY_IE_MISSING`:

الإلزامي IE	النوع
`nfConsumerIdentification`	كائن
`invocationTimeStamp`	سلسلة (ISO 8601)
`invocationSequenceNumber`	عدد صحيح

ChargingDataRequest: الحقول الرئيسية المستخدمة

الحقل	النوع	المستخدم في	وصف
`subscriberIdentifier`	سلسلة	الكل	SUPI (مثل `imsi-999700000000001`). `Account` في CGRat `nfConsumerIdenti` إذا كان غائبًا.
`notifyUri`	سلسلة	الإنشاء	زنة لإشعار الشحن URI لاحقًا (REAUTHORIZA ABORT_CHARGING)
`pDUSessionChargingInformation`	كائن	الكل	تفاصيل جلسة PDU: `pduSessionInforma` وع PDU، معرف جلسة `qoSInformation`، `networkSlicingIn`
`multipleUnitUsage`	مصفوفة	التحديث، التحرير	ام المبلغ عنها. يتم جمع إدخالات `usedUnitCo` /المدة المتراكمة. يحمل وعة التصنيف المطلوبة
`requestType`	سلسلة	الكل	`INITIAL_REQUEST`، `UPDATE_REQUEST`، و `TERMINATION_REQU`

الحقول الرئيسية المعادة: ChargingDataResponse

الحقل	النوع	الوصف
`invocationSequenceNumber`	عدد صحيح	عند الإنشاء؛ يتم زيادته في كل تحديث `1`.
`invocationTimeStamp`	سلسلة	طابع زمني ISO 8601 للاستجابة.
`multipleUnitInformation`	مصفوفة	الوحدات الممنوحة: إدخال واحد مع `resultCode: "SUCCESS"`، `grantedUnit` (`totalVolume`، `time`) ومجموعة التصنيف المح��ولة.
`triggers`	مصفوفة	محفزات إدارة الحصص التي تعلم المستهلك متى يعود: `QUOTA_THRESHOLD` عند نصف الحجم (IMMEDIATE_REPORT) و`TIME_LIMIT` (الممنوح عند 3600 ثانية DEFERRED_REPORT).

تكامل CGRateS

CGRateS هو محرك التصنيف والتوازن الخارجي. تعمل OmniCHF كعميل لجلسة **SessionS**، حيث تقوم بتعيين كل عملية شحن Nchf إلى طريقة JSON-RPC لجلسة:

عملية Nchf	طريقة CGRateS SessionS
الإنشاء (INITIAL)	`SessionSv1.InitiateSession`
التحديث (UPDATE)	`SessionSv1.UpdateSession`
التحرير (TERMINATE)	`SessionSv1.TerminateSession`
التحقق من الصحة	`SessionSv1.GetActiveSessions` (حد 1)

تُمرر قيمة `MaxUsage` التي تعيدها CGRateS مباشرة إلى المستهلك كـ `grantedUnit` (كلا من `totalVolume` و `time`).

السلوك الممكّن مقابل المعطّل

الجانب	`cgrates_enabled: true`	وضع `cgrates_enabled: false` (التجاوز)
تفويض الائتمان	بتصنيف CGRateS حقيقي: تقوم الحساب وإرجاع `MaxUsage`	لا شيء: يتم إرجاع منحة ثابتة قدرها **86400** وحدة لكل من الإنشاء والتحديث
تأثير الرصيد	يتم خصم/حجز رصيد المشترك في CGRateS	لا يتم لمس أي رصيد
إنهاء	ينهي `TerminateSession` جلسة CGRateS و CDR	لا شيء
التحقق من الصحة	تحقق حي ضد `cgrates_url`	دائمًا يُبلغ `reachable: false`، `cgrates_url: "disabled"`
توقيع السجل	`Initiating CGRateS session for ...`	`CGRateS integration disabled, returning default authorization` (تحذير)
حالة الاستخدام	الشحن عبر الإنترنت في الإنتاج	اختبار المعاملات / التكامل بدون محرك تصنيف

> في وضع التجاوز، لا تزال OmniCHF تخصص `chargingDataRef`s، وتتبع الجلسات، وتزيد أرقام التسلسل، وتصدر المحفزات، وتولد CDRs تم الذي هو الائتمان في التحكم فقط ؛ تخطيه. هذا يجعل من الآمن ممارسة التدفق الكامل SMF↔CHF دون نشر CGRateS.

تم توثيق تكوين تكامل CGRateS (`cgrates_enabled`، `cgrates_url`، `cgrates_tenant`، `cgrates_timeout`) في مرجع التكوين.

رسمG إلى CGRateS 5 أحداث

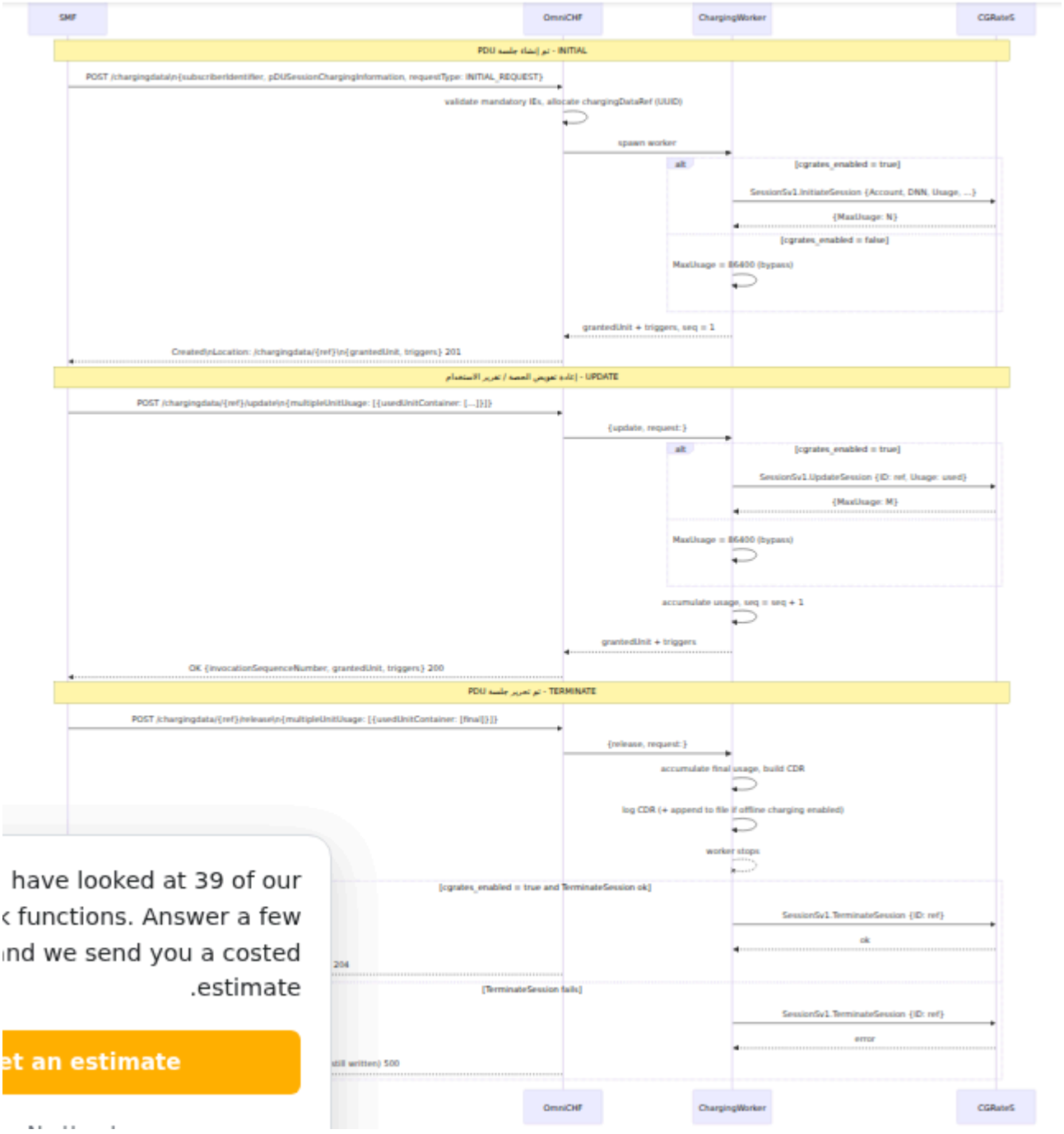
خريطة مبنية من `Event` يحمل كل استدعاء لجل��ة ChargingDataRequest:

حقل حدث CGRateS	المصدر	
`Account` / `Subject`	`subscriberIdentifier` (SUPI)	يعود إلى `nfConsumerI` أو `"unknown"`
`Destination` / `DNN`	`pduSessionInformation.dnnId` (أو `.dnn`)	م شبكة البيانات
`ToR`	مشتق من `pduType`	ت `*data` دائمًا
`RequestType`	`requestType`	`*` يتم تعيينه إلى
`Category`	ثابت	`"data"`.
`Usage`	`usedUnitContainer`	غير صفرية من `totalVolume` `downlinkVolu`
`OriginID`	`chargingDataRef`	فريد لكل جلسة
`OriginHost`	ثابت	`"OmniCHF"`.
`SUPI`	`subscriberIdentifier`	حقل امتداد 5G.
`S-NSSAI_SST` / `S-NSSAI_SD`	`pduSessionInformation.sNSSAI`	الافتراضات: SS
`5QI`	`qoSInformation.5qI` (أو `fiveQI`)	الافتراض `9`.
`RATType`	`pduSessionInformation.ratType`	الافتراض `"NR"`
`PDUSessionID`	`pduSessionInformation.pduSessionID`	الافتراض `0`.
`PDUSessionType`	`pduSessionInformation.pduType`	الافتراض `"IPV`

حقل حدث CGRateS	المصدر	
`ChargingOperation`	العملية	`Initial` / `Up`

الإجراءات الرئيسية

جلسة الشحن المتقاربة (INITIAL → UPDATE → TERMINATE)



have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate
et an estimate

تنهي وتكتب OmniCHF عند التحرير، لا تزال `TerminateSession` إذا فشلت CGRateS السبب) لذا لا يتم فقدان السجل غير المتصل. يتلقى المستهلك بعد ذلك `500`، CDR قد تظل مفتوحة وقد تحتاج CGRateS بدلاً من `204` زائف، لأن جلسة (`SYSTEM_FAILURE` الخلفية الفوترة إلى التسوية. عند إنهاء ناجح، يتلقى المسته��ك `204` `No Content`.

إدارة الحصص (RSU / GSU والمحفزات)

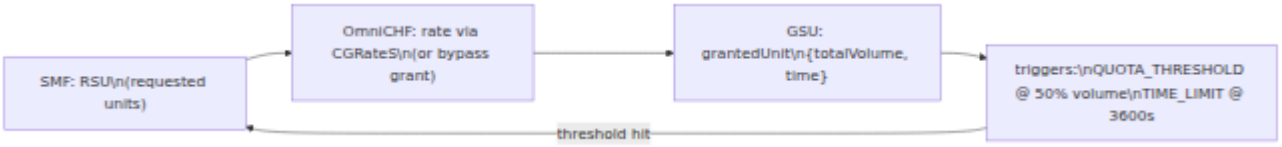
وحدات OmniCHF وتعيد (RSU) يرسل المستهلك وحدات الخدمة المطلوبة، TS 32.291 وفقًا لـ الخدمة الممنوحة (GSU) بالإضافة إلى **المحفزات** التي تخبر المستهلك متى يعود:



يعيد المستهلك التفويض (يرسل تحديثًا) عندما يصل إلى حد الحجم أو حد الوقت. يمكن أيضًا لـ OmniCHF دفع إعادة التفويض أو الإنهاء بشكل استباقي عبر نقطة النهاية `/notify` (`REAUTHORIZATION` / `ABORT_CHARGING`)، التي تنشر إلى `notifyUri` المقدم في الطلب الأولي.

إشعار الشحن (REAUTHORIZATION / ABORT_CHARGING)

وفقًا للفقرة 6.1.6.2 من TS 32.291، تدفع OmniCHF `ChargingNotifyRequest` إلى `notifyUri` المخزنة للمستهلك. المحفز هو نقطة النهاية التي تواجه المشغل `POST /nchf-convergedcharging/v3/notify`، التي يحمل جسمها `chargingDataRef` المستهدف الخاصة بها، وتنشر `notifyUri` عن الجلسة، وتحل OmniCHF تبحث. `notificationType`و الإشعار إلى المستهلك.



> الإشعار هو جهد أفضل نحو واجهة المشغل: يتم تسجيل `notifyUri` المرفوض أو غير القابل للوصول ولكن لا يغير نتيجة الجلسة التي تؤدي `notifyUri`. بمجرد حل الجلسة و `/notify` مخزنة إلى خطأ داخلي `no_notify_uri`. لا تحتوي على `notifyUri`

حقول CDR

يتم بناء CDR عند التحرير، ويتم تسجيله في INFO (`CHF CDR: %{...}`)، وعندما يكون `offline_charging_enabled` هو `true`، يتم إضافته كسطر JSON واحد إلى `cdr_YYYYMMDD.json`:

الحقل	المصدر
`record_type`	`5"` ثابت `G_PDU_SESSION"`
`supi`	سياق الجلسة
`dnn`	سياق الجلسة / `pduSessionInformation`
`snssai`	`{sst, sd}`
`qos_5qi`	`qoSInformation.5qI` (`9` الافتراض)
`rat_type`	`pduSessionInformation.ratType` (الافتراض `"NR"`)
`pdu_session_id` / `pdu_session_type`	سياق الجلسة / الطلب
`volume_uplink` / `volume_downlink` / `volume_total`	يتم جمعها من جميع `usedUnitContainer`s
`duration`	الوقت المبلغ عنه، وإلا الفرق الزمني
`start_time` / `end_time`	طوابع زمنية لإنشاء الجلسة / التحرير
`charging_data_ref`	UUID للجلسة
`charging_id`	معرف الشحن 32 بت من 3GPP (TS 32.251)
`recording_entity`	عنوان URL الأساسي لـ OmniCHF SBI
`rating_group`	تم حله من خريطة DNN

يتم تكوين مخرجات CDR غير المتصلة (`offline_charging_enabled`، `cdr_output_dir`) وتركيب معرف ا��شحن (`node_id`) في مرجع التكوين.

تشغيل الشحن غير المتصل

لإلتقاط CDRs الفوترة / إلى ملف للوساطة:

1. قم بتعيين `offline_charging_enabled: true` وأشر إلى `cdr_output_dir` إلى دليل قابل للكتابة.
2. تتراكم CDRs ككائن JSON واحد لكل سطر في ملف يومي يسمى `cdr_YYYYMMDD.json` (التاريخ هو تاريخ الإصدار UTC). يتم فتح ملف جديد كل يوم ؛ لا يوجد تدوير تلقائي أو احتفاظ يتجاوز تقسيم اسم الملف اليومي، لذا قم بإخراج الملفات القديمة بنفسك.
3. كل جلسة تم تحريرها تضيف بالضبط CDR واحد. تحتفظ الجلسات الجارية باستخدامها في العامل / السياق حتى التحرير؛ لا يتم كتابتها إلى الملف حتى ذلك الحين.

إذا كان `offline_charging_enabled` هو `false`، لا تزال CDRs تُبنى وتُسجل في INFO كـ `CHF CDR: %{...}`؛ يتم تخطي كتابة الملف فقط. لالتقاط CDRs بدون إخراج ملف، قم بشحن تلك الأسطر المسجلة مع خط أنابيب السجل الخاص بك. انظر CDRs مفقودة من الملفات غير المتصلة.

تحديد موقع والبحث عن CDRs

تقوم واجهة إدارة API بتعريض `GET /api/oam/cdr` للبحث في ملفات CDR المكتوبة. تقرأ كل ملف `cdr_*.json` تحت `cdr_output_dir` وتقوم بتصفية المعلمات الاستعلام المقدمة (جميعها اختيارية، مجمعة مع AND):

معلمة الاستعلام	تطابق	التنسيق
`supi`	CDR `supi` بالضبط	مثل `imsi-999700000000001`
`dnn`	CDR `dnn` بالضبط	مثل `internet`
`date_from`	تاريخ ملف CDR `>=` هذا التاريخ	`YYYY-MM-DD` (الشرطات اختيارية)
`date_to`	تاريخ ملف CDR `<=` هذا التاريخ	`YYYY-MM-DD` (الشرطات اختيارية)

تُطبق عوامل تصفية التاريخ على **تاريخ اسم الملف**، وليس على طوابع زمنية السجلات الفردية، لذا يتم تضمين أو استبعاد الملف بالكامل ليوم واحد كوحدة. بدون معلمات، ترجع نقطة النهاية كل

ترجع السجلات فقط `GET /api/oam/cdr` على القرص. نظرًا لأنها تقرأ الملفات، فإن CDR مفعلًا عند وقت الإصدار `offline_charging_enabled` عندما كان.

تفريغ CDR بالقوة (OAM)

`POST` فقط عندما يتم تحرير جلسة. يتيح CDR في التشغيل العادي، يتم إنتاج مبكرًا لأي جلسة حية (انظر تخزين الجلسة ورؤية `/api/oam/cdr/flush` لمشغل إصدار CDR الإدارة)، على سبيل المثال لإغلاق الجلسات التي لم يحررها المستهلك أبدًا:

جسم الطلب	التأثير
`{"charging_data_ref": "<ref>"}`	بناء وكتابة CDR لتلك الجلسة الواحدة، ثم إنهائها. إذا لم يتم العثور على المرجع `404`.
`{}` أو `{"charging_data_ref": "all"}`	تفريغ كل جلسة يتم تتبعها حاليًا.

يبني التفريغ CDR من الاستخدام المتراكم للجلسة بالإضافة إلى آخر `ChargingDataRequest` كالإفراج؛ لا يستدعي CDR ويعيد استخدام نفس كاتب **TerminateSession** من CGRateS. لكل جلسة. إذا كان `offline_charging_enabled` و `write_result` تقارير الاستجابة ويتم إنهاء الجلسة، ولكن لا يتم كت��بة أي شيء إلى CDR الشحن غير المتصل معطلاً، يتم بناء الملف (`write_result: "ok"` كل تفريغ يزيد من .(بدون إخراج ملف `omni_chf_charging_releases_total{result="oam_flush"}`.

← نظرة عامة

استكشاف أخطاء OmniCHF

400 طلب غير صحيح في أي طلب شحن

الأعراض: يتلقى المستهلك `400` مع السبب `MANDATORY_IE_MISSING`.

الأسباب المحتملة:

- جسم الطلب يفتقر إلى `nfConsumerIdentification`، `invocationTimeStamp`، أو `invocationSequenceNumber`.

الحل:

1. تأكد من أن المستهلك يتضمن جميع الثلاثة IEs الإلزامية في كل عملية (إنشاء، تحديث، إطلاق).
2. تحقق من حقل `detail` في استجابة ProblemDetails ؛ حيث يسرد أسماء IEs المفقودة.

404 عند التحديث أو الإطلاق

الأعراض: يتلقى المستهلك `404` مع السبب `CHARGING_DATA_NOT_FOUND`.

الأسباب المحتملة:

- تم إعادة تشغيل OmniCHF بين عملية الإنشاء والعملية اللاحقة؛ يتم الاحتفاظ بحالة الجلسة في الذاكرة، لذا يجب إعادة إنشاء الجلسة بعد إعادة التشغيل.
- `chargingDataRef` في المسار غير صحيح.
- تم إطلاق الجلسة بالفعل (توقف العامل الخاص بها).

الحل:

1. ابحث في السجلات عن `CHF Update: unknown ref=` / `CHF Release: unknown ref=`.
2. تأكد من أن OmniCHF لم يتم إعادة تشغيله منذ إنشاء الجلسة.
3. تأكد من أن المستهلك يعيد إرجاع `chargingDataRef` من رأس `Location` الخاص بعملية الإنشاء.

CGRateS عند الإنشاء مع تمكين 500

الأعراض: يتلقى المستهلك `500` مع السبب `CHARGING_FAILED`.

الأسباب المحتملة:

- فشل `InitiateSession` في CGRateS: محرك غير قابل للوصول، مستأجر خاطئ، أو خطأ من جانب CGRateS.

الحل:

1. تحقق من أن `cgrates_url` يشير إلى نقطة نهاية CGRateS `/jsonrpc` قابلة للوصول.
2. تحقق من أن `cgrates_tenant` يتطابق مع المستأجر في CGRateS.
3. تحقق من مقياس `omni_chf_cgrates_health` (`1` = نشط).
4. راجع السجلات للبحث عن `CGRateS InitiateSession failed ...`؛ السبب هو أحد `{:cgrates_error, msg}`، `{:http_error, status}`، و`{:http_error, reason}`.
5. تأكد من أن خدمة `SessionSv1` مفعلة في CGRateS.

500 الإطلاق يعيد

الأعراض: يتلقى المستهلك `500` مع السبب `SYSTEM_FAILURE` والعنوان "فشل إنهاء الشحن" على `POST /chargingdata/{ref}/release`.

الأسباب المحتملة:

- فشل `TerminateSession` في CGRateS أثناء الإطلاق (محرك غير قابل للوصول، مستأجر خاطئ، أو خطأ من جانب CGRateS).

التأثير:

- لا يزال يتم بناء CDR وكتابته، لذا فإن السجل غير المتصل ليس مفقودًا.
- قد تظل الجلسة من جانب CGRateS مفتوحة، لذا قد يكون هناك عدم تطابق في نظام الفوترة ويحتاج إلى تسوية.

الحل:

1. ابحث في السجلات عن `CGRateS TerminateSession failed` على `ref` المتأثر.
2. تأكد من إمكانية الوصول إلى CGRateS باستخدام مقياس `omni_chf_cgrates_health` (`1` = نشط).
3. قم بتسوية أي جلسة CGRateS تركت مفتوحة لذلك `ref` مقابل CDR المكتوب.
4. اربط مع `omni_chf_charging_releases_total{result="failure"}` لرؤية معدل فشل الإنهاء.

CGRateS فشل فحص صحة ينما�� الخدمة تعمل

الأعراض: يقرأ `omni_chf_cgrates_health` `0` ولكن CGRateS قيد التشغيل.

الأسباب المحتملة:

- CGRateS بطيء: يحدد فحص الصحة مهلة قصوى عند `min(cgrates_timeout, 3000)` ملي ثانية.
- عدم تطابق في البروتوكول (HTTP مقابل HTTPS) على `cgrates_url`.

الحل:

1. تأكد من أن بروتوكول `cgrates_url` والمنفذ صحيحان.
2. إذا كان CGRateS بطيئًا بشكل شرعي، تحقق من تحميل CGRateS بدلاً من رفع الحد الأقصى.
3. ابحث عن تحذيرات `CGRateS unreachable at <url> ...`.

CGRateS ارتفاع زمن الاستجابة في

الأعراض: عمليات شحن بطيئة؛ ارتفاع `omni_chf_cgrates_request_duration_ms`.

الحل:

1. افحص الهستوجرام p95/p99 حسب `operation`.
2. قم بتوسيع CGRateS أو مسار الشبكة بين OmniCHFوCGRateS.
3. فكر في تقليل `cgrates_timeout` حتى تفشل المكالمات البطيئة سريعًا بدلاً من حظر الجلسة.

CDRs مفقودة من الملفات غير المتصلة

الأعراض: تظهر خطوط `CHF CDR:` في السجل، ولكن لا يتم كتابة أي ملف `cdr_YYYYMMDD.json`.

الأسباب المحتملة:

- `offline_charging_enabled` هو `false` (الإعداد الافتراضي): يتم تسجيل CDRs فقط.
- `cdr_output_dir` غير قابل للكتابة.

الحل:

1. قم بتعيين `offline_charging_enabled: true`.
2. تأكد من أن `cdr_output_dir` موجود وقابل للكتابة؛ راقب الأخطاء `[CDR] Failed to write CDR to file`.
3. لالتقاط CDRs بدون إخراج ملفات، قم بشحن خطوط سجل `CHF CDR:` مع خط أنابيب السجل الخاص بك.

عدد الجلسات النشطة لا ينخفض

الأعراض: يبقى `omni_chf_sessions_active_count` مرتفعًا.

الأسباب المحتملة:

- لم يرسل المستهلك إطلاقًا (تتضح الجلسات فقط عند الإطلاق أو تعطل العامل).
- تعود الإطلاقات `404` (انظر أعلاه) لذا يعتبرها المستهلك مكتملة بينما قد لا تكون كذلك OmniCHF.

الحل:

1. اربط العدد النشط بمعدلات الإنشاء مقابل الإطلاق.
2. تحقق من `404` عند الإطلاق، والذي يشير عادةً إلى إعادة تشغيل سابقة.

غير محفوظ NRF تسجيل

الأعراض: يقرأ `omni_chf_nrf_registration_status` `0`.

الحل:

1. تحقق من أن `nrf_uri` قابل للوصول من `sbi_addr` الخاص بـ OmniCHF.
2. تأكد من أن `mcc` / `mnc` تتطابق مع تكوين PLMN الخاص بـ NRF.
3. تأكد من أن `sbi_addr` / `sbi_port` في الملف التعريفي المعلن قابلة للوصول من قبل NRF.
4. راجع سجلات بدء التشغيل للبحث عن أخطاء تسجيل NRF.