← نظرة عامة

مرجع تكوين OmniNSSF

يتم توفير تكوين وقت التشغيل عبر `:omnissf`. يتم قراءة تكوين المشغل من مفتاح بيئة التطبيق الذي يربط متغيرات بيئة نظام التشغيل بمفاتيح التكوين عند بدء `config/runtime.exs`، التشغيل. يتم توفير طوبولوجيا الشريحة (قائمة NSI، خريطة مجموعة NSSAI المسموح بها، AMF من خلال تكوين تطبيق هيكلي (انظر تكوين طوبولوجيا الشريحة) ويتم تعديله في (المكون NSSAI وقت التشغيل من خلال واجهة برمجة التطبيقات OAM.

تكوين وقت التشغيل (`config/runtime.exs`)

```
config :omnissf,
  sbi_scheme: System.get_env("SBI_SCHEME", "http"),
  sbi_addr: System.get_env("SBI_ADDR", "127.0.0.14"),
  sbi_port: String.to_integer(System.get_env("SBI_PORT", "7777")),
  nrf_uri: System.get_env("NRF_URI", "http://127.0.0.1:7777"),
  mcc: System.get_env("MCC", "999"),
  mnc: System.get_env("MNC", "70"),
  prometheus_metrics_port:
String.to_integer(System.get_env("PROMETHEUS_PORT", "9568")),
  heartbeat_interval:
String.to_integer(System.get_env("HEARTBEAT_INTERVAL", "10000"))

config :logger, :default_formatter,
  format: {OmniLogger.JsonFormatter, :format},
  metadata: :all
```

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`

ملاحظة: القيمة الافتراضية المجمعة لـ `nrf_uri` في تكوين وقت التشغيل هي `http://127.0.0.1:7777`، بينما القيمة الافتراضية للوصول الداخلي هي

> تكون القيمة الافتراضية الفعالة، `runtime.exs` عند تطبيق `http://127.0.0.10:7777`. قم بتعيين `NRF_URI` بشكل صريح لتجنب الغموض. هي `http://127.0.0.1:7777`.

إعدادات السلوك والإشعارات

تقوم مفاتيح تكوين تطبيق `:omnissf` التالية بضبط س��وك اختيار الشريحة والإشعارات. لا تحتوي أي منها على متغير بيئة `runtime.exs`، لذلك يتم تعيينها من خلال تكوين التطبيق بدلاً من متغير بيئة نظام التشغيل.

المعلمة	النوع	مطلوب	الافتراضي	
`dynamic_nrf_discovery`	Boolean	لا	`false`	ا تكون N يقوم م NRF ي وقت يل من ذر عبر Nnrf_N ستخدام `nsi_l` ل طلب شريحة، علم عن ل على S-NSS ود إلى `nsi_l` كتشاف متاح أو جع شيئًا
`strict_availability`	Boolean	لا	`false`	التوفر. ا تكون ،(مرن) ل توفر ال ينتج NSSAI ا تكون `true` ثل هذا وتُرجع ا فارغة
`nssf_notification_max_attempts`	عدد صحيح	لا	3	الأقصى لتسليم ير توفر

ـ	الافتراضي	مطلوب	النوع	المعلمة
سقاطه الفشل				
جع لكل ي ثانية ت تغيير تطبيق لى قبل الثانية، الثالثة، القيمة حاولات إضافية	`[200, 1000]`	لا	قائمة من الأعداد الصحيحة	`nssf_notification_backoff_ms`

السجل

وصف	متغير البيئة	الافتراضي	مطلوب	النوع	المعلمة
تنسيق سجل JSON يكلي سجل اضي	-	`{OmniLogger.JsonFormatter, :format}`	لا	مجموعة تنسيق	`format`
بانات عريف سجلة ضمنة ي كل دخال (`:al`) ر كل حقل بانات عريف	-	`:all`	لا	ذرة / قائمة	`metadata`

يمكن تغيير مستوى السجل النشط في وقت التشغيل دون إعادة تشغيل من خلال واجهة برمجة التطبيقات OAM (`PATCH /oam/config/{id}` مع `log_level`، أو `POST /oam/log_level`). المستويات الصالحة هي `emergency`، `alert`، `critical`، `error`، `warning`، `warn`، `notice`، `info`، و `debug`.

تكوين طوبولوجيا الشريحة

عند بدء التشغيل من مفاتيح تكوين `SliceConfig` يتم تحميل طوبولوجيا الشريحة في مخزن واحدة eMBB التالية. يحتوي كل منها على قيمة افتراضية مدمجة تمثل شريحة تطبيق `:omnissf` (SST 1، SD `0x000001`). هذه المفاتيح هي هيكلية (خرائط وقوائم) ويتم تعيينها من خلال تكوين.

التطبيق بدلاً من متغير بيئة واحد؛ يمكن أيضًا تغييرها في وقت التشغيل من خلال واجهة برمجة التطبيقات OAM (انظر إدارة الشريحة في وقت التشغيل).

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 1, sd: "0x000001"}, nrf_uri:
"http://127.0.0.10:7777", nsi_id: "1"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 1, sd: "0x000001"}]
  },
  amf_set_mapping: %{
    "1-0x000001" => ["1"]
  },
  configured_nssai: %{}
```

`nsi_list` - إدخالات مثيل شريحة الشبكة

يربط كل إدخال S-NSSAI بـ NRF الذي يخدم اكتشاف NF المحدد بالشريحة لتلك الشريحة. يقوم NSSF بمطابقة S-NSSAI التي يتم NSI المطلوب ضد هذه القائمة؛ تطابق يؤدي إلى معلومات إرجاعها إلى AMF، `403` وعدم المطابقة يؤدي إلى.

المعلمة	النوع	مطلوب	الافتراضي	متغير البيئة	
`nsi_list`	قائمة من الخرائط	لا	إدخال eMBB (أدناه) واحد	-	لات NS غيل
`s_nssai.sst`	عدد صحيح	نعم	`1`	-	دمة يحة
`s_nssai.sd`	سلسلة	لا	`"0x000001"`	-	فه، أي تلك
`nrf_uri`	سلسلة	نعم	`"http://127.0.0.10:7777"`	-	UR يتم إلى `ns`
`nsi_id`	سلسلة	لا	`"1"`	-	كة، ه كـ `ns` عند يينه

على أي من الإدخال المكون أ❓❓ الطلب يتطابق مع `nil` SD: مدرك للرموز البديلة SD تطابق أي SD، وSDs غير `nil` تتم مقارنتها بدون اعتبار لحالة الأحرف.

`allowed_nssai` - NSSAI المسموح بها لكل PLMN

يربط مفتاح PLMN `"mcc-mnc"` بقائمة S-NSSAIs المسموح بها لـ UEs من ذلك PLMN. عندما لا تحتوي PLMN يعود، بالاستعلام على إدخال الخاصة NSSF إلى PLMN المنزل (`mcc-mnc` من تكوين وقت التشغيل). يتم إرجاع القائمة الناتجة كـ `allowedNssaiList` (نوع الوصول `3GPP_ACCESS`) وعندما يتم تقديم TAI، يتم تصفيتها إلى S-NSSAIs المبلغ عنها المتاحة في تلك TAI.

المعلمة	النوع	مطلوب	الافتراضي	متغير البيئة	الوصف
`allowed_nssai`	خريطة	لا	`{"999-70" => [...]}`	-	جدول NSSAI المسموح بها المعتمد على PLMN
مفتاح الخريطة	سلسلة `"mcc-mnc"`	نعم	`"999-70"`	-	معرف PLMN الذي ينطبق عليه مجموعة المسموح بها
`sst`	عدد صحيح	نعم	`1`	-	SST لـ S-NSSAI المسموح بها
`sd`	سلسلة	لا	`"0x000001"`	-	SD لـ S-NSSAI المسموح بها

`amf_set_mapping` - مجموعة AMF المستهدفة لكل S-NSSAI

يربط مفتاح S-NSSAI بمجموعة AMF (أو مجموعات) قادرة على خدمة تلك الشريحة. عند وجودها، `targetAmfSet` الأولى كـ AMF يتم إرجاع مجموعة بإعادة التوجيه إذا لم يتمكن AMF مما يخبر، من خدمة الشريحة بنفسه.

الوصف	متغير البيئة	الافتراضي	مطلوب	النوع	المعلمة
جدول مجموعة AMF المستهدفة المعتمد على مفتاح S-NSSAI	-	`{"1-0x000001" => ["1"]}`	لا	خريطة	`amf_set_mapping`
مفتاح S-NSSAI الذي ينطبق عليه التعيين	-	`"1-0x000001"`	نعم	سلسلة `"sst"` أو `"sst-sd"`	مفتاح الخريطة
معرفات مجموعة AMF المستهدفة للشريحة	-	`["1"]`	نعم	قائمة من السلاسل	القيمة

> هو **دقيق**، على عكس `configured_nssai` و `amf_set_mapping` تطابق المفاتيح لـ `nsi_list`. يتم بناء مفتاح البحث من S-NSSAI في الاستعلام: `"sst"` عندما لا تحتوي S-NSSAI المطلوبة على SD، و `"sst-sd"` الاستعلام الذي يحمل SD عندما تحتوي. يتطابق العاري لن يتطابق معه، والعكس صحيح. لا `"sst"` ؛ مفتاح `"sst-sd"` فقط مع م❓❓تاح SD. يوجد هنا تطابق برمجي لـ `"sst-sd"` استخدم شكل SD، إذا كنت تخدم شريحة مع SD، المطلوبة لا تحتوي على S-NSSAI ؛ إذا كانت(`x000001"` على سبيل المثال، `0-1"`) استخدم الشكل العاري `"sst"` (`1"` على سبيل المثال، `"`).

`configured_nssai` - تعيين NSSAI (التجوال) المكون

في سيناريوهات التجوال / AMF المكونة التي يتم إرجاعها إلى NSSAI بقيمة S-NSSAI يربط مفتاح التعيين. فارغ بشكل افتراضي؛ عند تعيينه، يتم إرجاع القيمة كما هي كـ `configuredNssai`. يتم استخدام ذلك لإعطاء UE المتجول التعيين بين S-NSSAI لشبكة PLMN و (المزارة) المقدمة S-NSSAI لشبكتها المنزلية عندما تنتقل S-NSSAI تقديم UE المقابل لشبكتها المنزلية، بحيث يمكن لـ NSSAI

بتفسير القيمة - بل يمرر أي كائن تقوم بتكوينه - لذا يجب أن يكون مشكلاً NSSF لاحقًا. لا يقوم TS 29.531. وفقًا لـ AMF تمامًا كما يتوقع

الوصف	متغير البيئة	الافتراضي	مطلوب	النوع	المعلمة
جدول NSSAI مكون المعتمد على مفتاح S-NSSAI	-	(فارغ) `%{}`	لا	خريطة	`configured_nssai`
ي مفتاح S-NSSAI ينطبق عليه التعيين المكون (مطابقة قة، انظر الملاحظة أعلاه)	-	(لا شيء)	لا	سلسلة `"sst"` أو `"sst-sd"`	مفتاح الخريطة
كونة NSSAI قيمة ني يتم إرجاعها كما هي كـ `configuredNss`	-	(لا شيء)	لا	كائن	القيمة

مثال - إرجاع تعيين S-NSSAI لشبكة PLMN `0-1` المنزلية للشريحة المزارة `x000001`:

```
config :omnissf,
  configured_nssai: %{
    "1-0x000001" => %{"sst" => 1, "sd" => "0x0000ff"}
  }
```

`configuredNssai` فارغًا (`%{}`) لنشر غير التجوال؛ ثم يتم حذف `configured_nssai` اترك تمامًا من الاستجابة.

تعريف شريحة من البداية إلى النهاية

لا يتم وصف شريحة واحدة بواسطة مفتاح تكوين واحد - المفاتيح الأربعة أعلاه هي جداول مستقلة، ويتم تجميع استجابة اختيار الشريحة من أي منها تحتوي على إدخال م��ابق. لجعل شريحة جديدة قابلة للاستخدام بالكامل، عادةً ما تقوم بتعبئة اثنين، وخياريًا جميع الأربعة:

المفتاح	مطلوب لجعل الشريحة تعمل؟	ما يتحكم فيه في الاستجابة
`nsi_list`	**نعم**	تحل على الإطلاق. يتم S-NSSAI ما إذا كانت NSI رفض الطلب الذي لا يحتوي على إدخال `SNSSAI_NOT_FOUND`. يوفر `403` مطابق بـ `nsiInformation` (NRF المستهدف + معرف NSI اختياري)
`allowed_nssai`	**نعم**، لكل PLMN خادم	أنه يمكنه استخدام UE ما إذا كان يتم إخبار ودون تراجع) مطابق PLMN الشريحة. بدون إدخال يتم حذف (المنزل PLMN إلى من الاستجابة، لذا لا يوجد `allowedNssaiList` UE لـ AMF مسموح به ليقدمه NSSAI
`amf_set_mapping`	اختياري	من AMF حتى يتمكن `targetAmfSet` يضيف إعادة التوجيه عندما لا يمكنه خدمة الشريحة واحدة AMF بنفسه. احذف لنشر مجموعة
`configured_nssai`	اختياري (التجوال فقط)	من S-NSSAI لتعيين `configuredNssai` يضيف المزارة إلى المنزل

مثال عملي - إضافة شريحة SST `2` (URLLC)، SD `0x000010` مقدمة من NRF، المنزلية `70-999`، قابلة لإعادة التوجيه PLMN مسموح بها لشبكة `http://10.0.0.20:7777`، إلى مجموعة AMF `2`:

```
config :omnissf,
  nsi_list: [
    %{s_nssai: %{sst: 2, sd: "0x000010"}, nrf_uri:
"http://10.0.0.20:7777", nsi_id: "2"}
  ],
  allowed_nssai: %{
    "999-70" => [%{sst: 2, sd: "0x000010"}]
  },
  amf_set_mapping: %{
    "2-0x000010" => ["2"]
  }
```

تذكر عدم التماثل في مطابقة SD: تقارن `nsi_list` و `allowed_nssai` SDs بدون اعتبار على) متسق SD دقيقة. حافظ على شكل `"sst-sd"` بواسطة سلسلة `configured_nssai` و `amf_set_mapping` يعمل كرمز بديل، بينما يتم تعيين `nil` لحالة الأحرف مع كون سبيل المثال دائمًا `"0x000010"`) عبر جميع المفاتيح الأربعة حتى تتماشى المفاتيح.

يمكن أيضًا إضافة نفس الشريحة في وقت التشغيل باستخدام `POST /oam/slice` (الذي يكتب جداول(`nsi_list`) فقط ؛ NSSAI المسموح بها، مجموعة AMF، و NSSAI المكونة هي تكوين NSI في وقت تتراجع إضافات .OAM فقط ولا يمكن تعديلها من خلال واجهة برمجة التطبيقات التشغيل عند إعادة التشغيل.

منافذ الإدارة والمقاييس

تبدأ المستمعات التالية الموجهة للمشغل جنبًا إلى جنب مع نقطة نهاية SBI:

الوصف	متغير البيئة	الافتراضي	المستمع
واجهة برمجة التطبيقات للحالة، فحص REST الشريحة، الإحصائيات، في وقت OAM وإجراءات التشغيل	-	TCP `8443` (HTTPS)	إدارة / واجهة برمجة التطبيقات OAM
نقطة نهاية تجريف Prometheus تعرض مقاييس BEAM و NSSF	`PROMETHEUS_PORT`	TCP `9568`	مقاييس Prometheus

---

إدارة / واجهة برمجة التطبيقات OAM

تُقدم واجهة برمجة التطبيقات REST للإدارة / OAM عبر HTTPS (ترتبط بـ `8443` على المنفذ `0.0.0.0` في وقت OAM). توفر حالة وإحصائيات للقراءة فقط بالإضافة إلى عناصر تحكم التشغيل. جميع الاستجابات هي JSON، مغلفة في ظرف قياسي `success` / `error`. المسارات أدناه نسبية إلى قاعدة واجهة برمجة التطبيقات.

الحالة والفحص

الطريقة	المسار	الوصف
GET	`/status/api`	حيوية خدمة واجهة برمجة التطبيقات
GET	`/status/nrf`	حالة تسجيل NRF
GET	`/status/nf`	حالة NF لـ NSSF
GET	`/slice_config`	لقطة تكوين الشريحة الحالية
GET	`/api/slices`	NSI، المكونة مع تعيينات S-NSSAIs المستهدفة، NSSAI مجموعات AMF، وتعيين مجموعة PLMN المسموح بها لكل AMF
GET	`/api/nssai_availability`	توفر NSSAI لكل AMF المخزنة حاليًا
GET	`/api/statistics`	إحصائيات تشغيل NSSF
GET	`/oam/config`	التكوين الفعلي في وقت التشغيل (/عنوان سعة، MCC/MNC، URI NRF، SBI منفذ، مستوى السجل/NF، أولوية)

لقطة العد مفيدة لفحص سريع للصحة / السعة دون تجريف `GET /api/statistics` تُرجع Prometheus:

الحقل	المعنى
`total_slices`	عدد إدخالات NSI المحملة حاليًا (حجم `nsi_list`)
`total_allowed_nssai_plmns`	عدد PLMNs مع إدخال `allowed_nssai`
`total_amf_set_mappings`	عدد مفاتيح S-NSSAI في `amf_set_mapping`
`nssai_availability_count`	عدد AMFs التي أبلغت عن توفر NSSAI لكل TAI
`nssai_availability_subscription_count`	عدد الاشتراكات النشطة لتغيير التوفر

إدارة الشريحة في وقت التشغيل

الطريقة	المسار	الوصف
POST	`/oam/slice`	إضافة أو تحديث تعيين NSI. الجسم: `sst` (مطلوب)، `nrf_uri` (مطلوب)، `sd` اختياري، `nsi_id`
DELETE	`/oam/slice/{id}`	إزالة تعيين NSI. `id` هو مفتاح S-NSSAI `"sst"` أو `"sst-sd"`. تُرجع `404` إذا لم يكن هناك NSI مطابق
DELETE	`/oam/nssai_availability/{id}`	إبطال توفر NSSAI المخزنة لـ AMF (`id` = `nfId`)

تطبق تغييرات الشريحة في وقت التشغيل على الفور على استعلامات اختيار الشريحة التالية ولكن لا يتم الاحتفاظ بها؛ تتراجع إلى الطوبولوجيا المكونة عند إعادة التشغيل.

عناصر التحكم في تكوين وقت التشغيل

الطريقة	المسار	الوصف
PATCH	`/oam/config/{id}`	تحديث تكوين وقت التشغيل. المفاتيح المسموح بها: `nrf_uri`، `nf_capacity`، `nf_priority`، `log_level`. المفاتيح / القيم غير الصالحة تُرجع `400`
POST	`/oam/nrf/reregister`	فرض إعادة تسجيل NRF
POST	`/oam/log_level`	تغيير مستوى السجل. الجسم: `level` (واحد من `emergency`، `alert`، `critical`، `error`، `warning`، `warn`، `notice`، `info`، `debug`)

`nf_priority` و `nf_capacity` يجب أن تكون، بالنسبة لـ `PATCH /oam/config/{id}`، سلسلة مستوى صالحة؛ خلاف ذلك، تُرجع الطلب `log_level` أعداد صحيحة ويجب أن يكون `400`.

← نظرة عامة

مقاييس OmniNSSF

تُعرض المقاييس بتنسيق Prometheus على منفذ المقاييس (`9568` الافتراضي، `PROMETHEUS_PORT`). المقاييس الخاصة بـ NSSF مقاييس تصدير يتم كما ؛أدناه مدرجة BEAM VM / الذاكرة (`beam.memory.*`، `beam.processes.count`، `beam.ports.count`، `beam.atom.count`، `beam.vm.uptime`).

مقاييس NSSF

المقياس	النوع	سميات
`omni_nssf_nsselection_requests_count`	عداد	`resul` `nf_ty`
`omni_nssf_ns_selection_requests_total`	عداد	`resul`
`omni_nssf_nssai_availability_update_count`	عداد	`nf_id`
`omni_nssf_nssai_availability_delete_count`	عداد	`nf_id`
`omni_nssf_nssai_availability_subscribe_count`	عداد	شيء)
`omni_nssf_nssai_availability_unsubscribe_count`	عداد	شيء)
`omni_nssf_nssai_availability_updates_total`	عداد	`nf_id`
`omni_nssf_nssai_availability_patch_total`	عداد	`nf_id`
`omni_nssf_nssai_availability_patch_subscription_total`	عداد	شيء)
`omni_nssf_nrf_registration_status`	مقياس	`nf_ty`

> يوجد عدادان متطابقان تقريبًا لاختيار الشريحة ولا ينبغي جمعهما معًا. يستخدم `omni_nssf_nsselection_requests_count` قيم `result` `success`/`error` ويحمل `omni_nssf_ns_selection_requests_total` قيم `nf_type`، بينما يستخدم أيضًا `result` `success`/`failure` ولا يحمل `nf_type`. اختر واحدًا كمصدر للحقيقة لتجنب العد المزدوج لنفس الطلبات.

`omni_nssf_nssai_availability_updates_total` (جمع `updates`) و `omni_nssf_nssai_availability_update_count` (مفرد) كلاهما متصل بنفس حدث تحديث PUT، لذا يتتبعان معًا. يمكن استخدام أي منهما لحجم تحديث التوفر الإجمالي.

مقاييس BEAM VM

المقياس	النوع	الوصف
`beam_memory_total`	مقياس	إجمالي الذاكرة المخصصة من قبل VM (بايت)
`beam_memory_processes` / `beam_memory_processes_used`	مقياس	الذاكرة المخصصة لـ / المستخدمة من قبل العمليات (بايت)
`beam_memory_system`	مقياس	الذاكرة المخصصة لنظام VM (بايت)
`beam_memory_atom` / `beam_memory_atom_used`	مقياس	الذاكرة المخصصة لـ / المستخدمة من قبل جدول الذرات (بايت)
`beam_memory_binary`	مقياس	الذاكرة المخصصة للثنائيات (بايت)
`beam_memory_code`	مق��اس	الذاكرة المخصصة للكود المحمل (بايت)
`beam_memory_ets`	مقياس	الذاكرة المخصصة لجداول ETS (بايت)
`beam_processes_count`	مقياس	عدد العمليات الحية
`beam_ports_count`	مقياس	عدد المنافذ المفتوحة
`beam_atom_count`	مقياس	عدد الذرات في جدول الذرات
`beam_vm_uptime`	مقياس	وقت تشغيل VM (ثواني)

مثال على PromQL

معدل خطأ اختيار الشريحة على مدى 5 دقائق:

```
sum(rate(omni_nssf_nsselection_requests_count{result="error"}
[5m]))
  / sum(rate(omni_nssf_nsselection_requests_count[5m]))
```

معدل طلب اختيار الشريحة حسب نوع NF المطلوب:

```
sum by (nf_type) (rate(omni_nssf_nsselection_requests_count[5m]))
```

صحة تسجيل NRF (تنبيه عند عدم التسجيل):

```
min(omni_nssf_nrf_registration_status) < 1
```

معدل تحديث توفر NSSAI حسب AMF:

```
sum by (nf_id)
(rate(omni_nssf_nssai_availability_update_count[5m]))
```

← نظرة عامة

دليل عمليات OmniNSSF

تقوم OmniNSSF بتنفيذ وظيفة اختيار شريحة الشبكة (NSSF) 5 في النواة 5G. إنها تحل إلى استجابة اختيار شريحة مصرح بها، وتجمع وتصرح بتوافر AMF استفسارات اختيار الشريحة من الشرائح لكل منطقة تتبع يتم الإبلاغ عنها بواسطة AMFs.

يتم الاحتفاظ بطوبولوجيا الشريحة (تعيينات S-NSSAI إلى NRF، NSSAI المسموح بها لكل PLMN، في الذاكرة يتم تحميله من `SliceConfig` في مخزن (المكونة NSSAIو AMF، تعيينات مجموعة AMF. يتم تكوين التطبيق عند بدء التشغيل ويمكن تعديله في وقت التشغيل من خلال واجهة إدارة OAM. الاحتفاظ بتوافر NSSAI لكل AMF واشتراكات التوافر في مخزن `Context` في الذاكرة. هذه هي الحية بعد إعادة التشغيل AMF حالة وقت التشغيل: يتم إعادة بنائها من التكوين ومن تقارير.

عند بدء التشغيل وتحافظ على تسجيلها مع نبضات قلب دورية. بشكل NRF مع OmniNSSF تسجل ويتم `nsi_list` المحددة للشريحة بواسطة المشغل في NRF افتراضي، يتم تكوين أهداف الديناميكي كخيار اختياري: عند تمكين NRF كما يتوفر اكتشاف NSI. كمعلومات AMF إرجاعها إلى `dynamic_nrf_discovery`، تقوم NSSF بحل NRF في كل طلب اختيار شريحة، الذي يخدم NRF الجذر عبر Nnrf_NFDiscovery، والعودة إلى `nsi_list` الشريحة في وقت التشغيل من المكونة مسبقًا عندما يكون الاكتشاف غير متاح أو لا يعطي أي شيء. يتم تعطيل هذا بشكل افتراضي.

ومراجع المواصفات 3GPP دور

المواصفة	الأهمية
TS 23.501 §5.15	مفاهيم تقسيم الشبكة - S-NSSAI، أنواع NSSAI (مطلوبة / مسموح بها /)، دور NSSF مثيل شريحة الشبكة، مشترك بها (مكونة /
TS 23.502	الإجراءات - اختيار الشريحة أثناء التسجيل وتأسيس جلسة PDU
TS 29.531	واجهات خدمة Nnssf_NSSelection و Nnssf_NSSAIAvailability (HTTP/2 SBI)
TS 23.003	تنسيق S-NSSAI - SST (1 بايت) و SD (3 بايت) الاختياري

مفاهيم S-NSSAI و NSSAI

معلومات مساعدة اختيار شريحة الشبكة الفردية (S-NSSAI) تحدد شريحة شبكة واحدة. وفقًا لـ TS 23.003، تتكون من:

- **SST (نوع الشريحة / الخدمة)** - `1` = إلزامي، قيمة 8 بت. تشمل القيم القياسية eMBB، `2` = URLLC، `3` = MIoT، `4` = V2X.
- **SD (مميز الشريحة)** - اختياري، قيمة 24 بت تميز بين شرائح متعددة من نفس SST. عندما يكون SD `0` أو `x000001` تُكتب عادة كسلسلة مكونة من 6 أرقام سداسية مثل `000001`. غائبًا، ينطبق S-NSSAI على أي/كل المميزين لتلك SST.

NSSAI هو مجموعة من S-NSSAIs. 5G يميز نظام عدة أنواع من NSSAI، جميعها تعمل معها NSSF:

نوع NSSAI	المعنى	أين تستخدم OmniNSSF
NSSAI المطلوب	S-NSSAIs التي يطلبها UE للاستخدام، يتم إرسالها بواسطة AMF في استفسار اختيار الشريحة	الإدخال - مستخرج من طلب معلومات الشريحة للتسجيل أو جلسة PDU
NSSAI المشترك	S-NSSAIs التي يحق لـ UE استخدامها وفقًا لاش��راكه	الإدخال - احتياطي عند عدم وجود مطلوب في استفسار NSSAI التسجيل
NSSAI المسموح بها	S-NSSAIs التي يمكن لـ UE استخدامها فعليًا في PLMN/المنطقة الخادمة	الإخراج - محسوبة من مجموعة PLMN المسموح بها لكل ومفلترة حسب توافر TAI
NSSAI المكونة	S-NSSAIs المكونة لـ UE في PLMN (تستخدم للتجوال/التعيين)	الإخراج - تُرجع عندما يتم تعريف S-NSSAI مكونة لـ NSSAI تعيين

المطلوب ضد قائمة **مثيل شريحة الشبكة** S-NSSAI استفسارًا كما يلي: تطابق NSSF تحل الخادم PLMN المسموح بها لـ NSSAI المحدد للشريحة، وتحسب NRF المكونة للعثور على **(NSI)** NSSAI المستهدفة وتعيين AMF وتضيف مجموعة، (المبلغ عنه TAI مفلترة اختياريًا حسب توافر) المكونة عندما تكون تلك مكونة للشريحة.

نقاط نهاية SBI

تتم خدمة جميع نقاط نهاية SBI عبر عنوان URL الأساسي `{sbi_scheme}://{sbi_addr}:{sbi_port}` (الافتراضي `http://127.0.0.14:7777`).

Nnssf_NSSelection (TS 29.531)

الطريقة	ال��سار	الوصف
GET	`/nnssf-nsselection/v2/network-slice-information`	تُرجع كائن AMF. حل اختيار الشريحة لـ `AuthorizedNetworkSliceInfo`

معلمات الاستعلام الإلزامية: `nf-type`, `nf-id`. مع السبب `400` تُرجع معلمة مفقودة إلزامية `MANDATORY_IE_MISSING`. يتم توفير إدخال الشريحة من خلال أحد `slice-info-request-for-pdu-session`، `slice-info-request-for-registration`، أو معلمة مباشرة `snssai`. تشمل المعلمات الاختيارية `tai`، `home-plmn-id`، و `supported-features`.

رموز الاستجابة:

الرمز	المعنى
`200`	`AuthorizedNetworkSliceInfo` (معلومات NSI، وعند الاقتضاء NSSAI المكونة، NSSAI المستهدفة، AMF المسموح بها، مجموعة)
`400`	معلمة إلزامية مفقودة أو لم يتم توفير معلومات الشريحة
`403`	لا يوجد NSI مكونة لـ S-NSSAI المطلوب (السبب `SNSSAI_NOT_FOUND`)

Nnssf_NSSAIAvailability (TS 29.531)

الطريقة	المسار	
PUT	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	S-NS تخزين قائمة من AMF. تُرجع `AuthorizedNss`
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	المخزن باستخدام JSON Patch (RF `application/j`
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/{nfId}`	التوافر المخزن لـ
POST	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions`	افر. تُرجع `201` مع رأس `Location`
PATCH	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	اشتراك باستخدام (RFC 6902)
DELETE	`/nnssf-nssaiavailability/v1/nssai-availability/subscriptions/{subscriptionId}`	شتراك. تُرجع `204`

على مصفوفة PUT/PATCH يجب أن يحتوي جسم توافر `supportedNssaiAvailabilityData` غير فارغة، تحمل كل إدخال كائن `tai` وقائمة على التوافر المخزن (أو على اشتراك) الذي PATCH غير فارغة. تُرجع `supportedSnssaiList` يشير إلى `{nfId}` / `{subscriptionId}` يجب أن يحتوي جسم الاشتراك. `404` غير معروف على `nfNssaiAvailabilityUri`، قائمة `taiList` غير فارغة، وحدث `event`. أي مسار SBI غير متطابق يُرجع `404`.

شكل بيانات التوافر ومطابقة TAI

كل إدخال `supportedNssaiAvailabilityData` له جزئين مطلوبين:

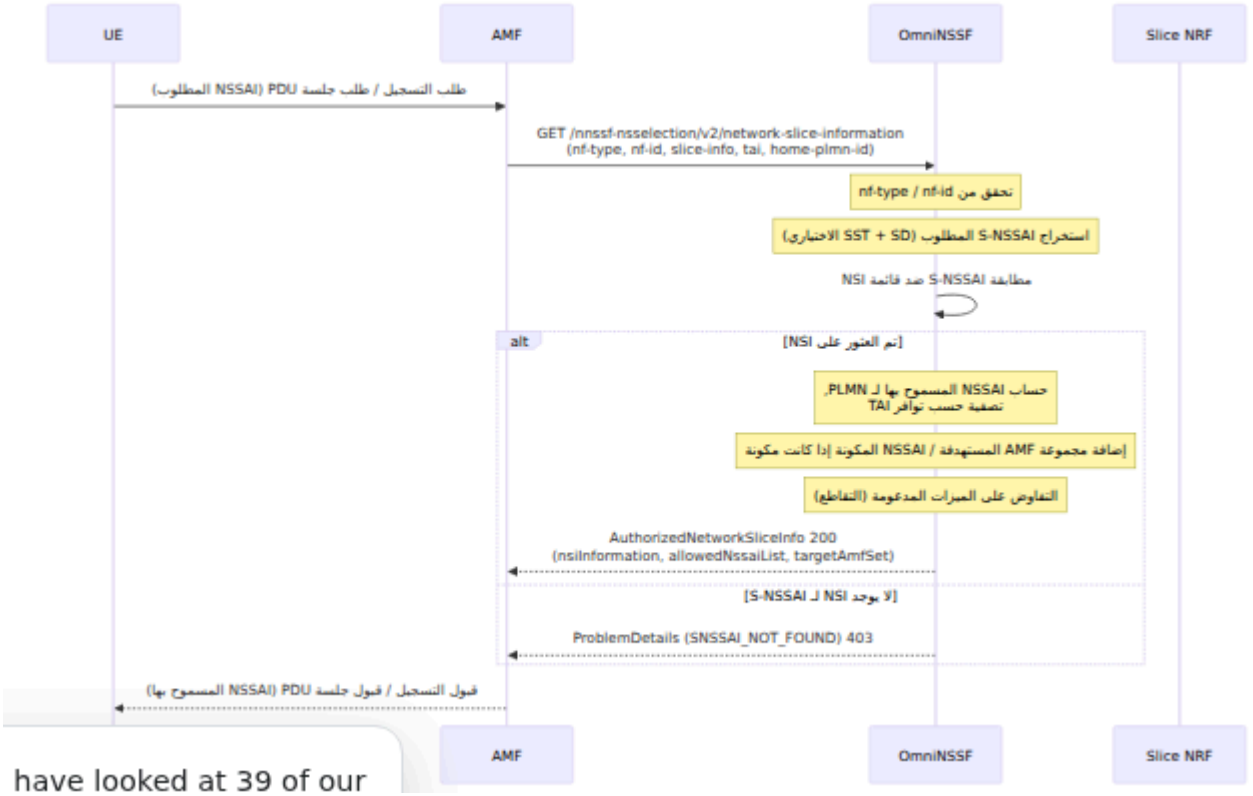
الحقل	الشكل	المعنى
`tai`	`{ "plmnId": { "mcc", "mnc" }, "tac" }`	منطقة التتبع التي ينطبق عليها التوافر
`supportedSnssaiList`	مصفوفة غير فارغة من `{ "sst", "sd"? }`	S-NSSAIs التي يدعمها AMF المبلغ في تلك TAI

استفسار (انظر TAI المسموح بها بواسطة NSSAI عندما تقوم عملية اختيار الشريحة لاحقًا بتصفية ويعتبر ، **`tac` و ، `mnc` ، `mcc`** اختيار شريحة الشبكة)، يكون المطابقة **دقيقة على جميع الثلاثة** مقارنة بدقة، أو SD) تدرجها TAI متاحًا فقط إذا كانت بعض الإدخالات المبلغ عنها لتلك S-NSSAI يمكن أن تظهر الشريحة غير متاحة ببساطة لأن .(SD عندما يغفل الاستفسار SD مطابقة أي تبلغ AMF عن ذلك الموجود في استفسار الاختيار - تأكد من أن PLMN أو `tac` يختلف TAI أبلغ عن التي ترمزها لاحقًا TAI عن نفس.

الإجراءات الرئيسية

اختيار شريحة الشبكة (Nnssf_NSSelection)

بحل NSSF تقوم. PDU وأثناء تأسيس جلسة UE أثناء تسجيل AMF يتم الاستفسار عنها بواسطة المطلوب إلى استجابة اختيار شريحة مصرح بها S-NSSAI.



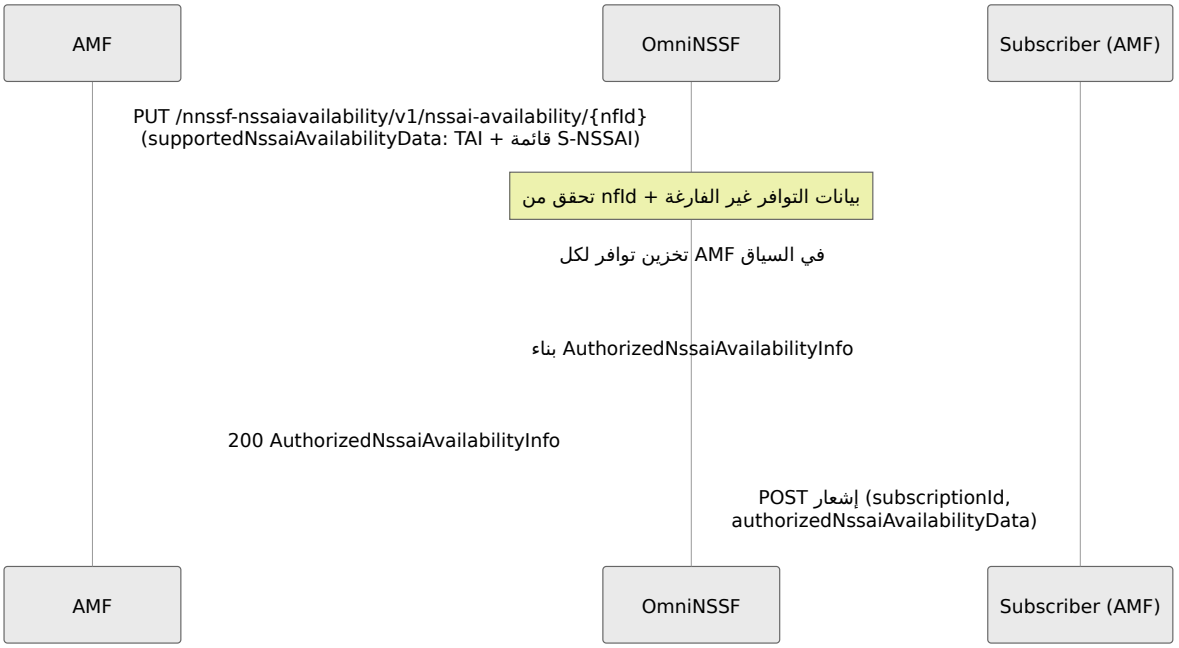
have looked at 39 of our

NSSF عن التوافر، تتحقق AMFs في الاستفسار وقد أبلغت واحدة أو أكثر من TAI عندما يتم توفير المسموح بها وفقًا لذلك. إذا لم يتم NSSAI وتصفية TAI المطلوب متاح في تلك S-NSSAI من أن افتراضي متساهل؛ قم بتعيين) الإبلاغ عن أي توافر بعد، يتم اعتبار جميع الشرائح المكونة متاحة `strict_availability: true` للفشل في الإغلاق). عندما يقدم AMF قيمة `supported-features`، تعيد NSSF حاليًا بت) التقاطع المتفاوض عليه ضد مجموعة ميزاتها الخاصة NssaiAvailability).

المطابق. عندما `nsi_list` عادةً من إدخال `nsiInformation` المستهدف المعاد كـ NRF يأتي الذي يخدم الشريحة NRF بدلاً من ذلك بحل NSSF تقوم، `dynamic_nrf_discovery` يتم تمكين S-NSSAI لـ Nnrf_NFDiscovery الجذر عبر NRF في وقت التشغيل عن طريق استفسار عندم�ا يكون الاكتشاف معطلاً أو `nsi_list` المكونة مسبقًا في NRF المطلوب، وتعود إلى غير متاح أو لا يُرجع أي شيء. انظر مرجع التكوين لمفتاح التكوين.

تحديث توافر NSSAI (Nnssf_NSSAIAvailability)

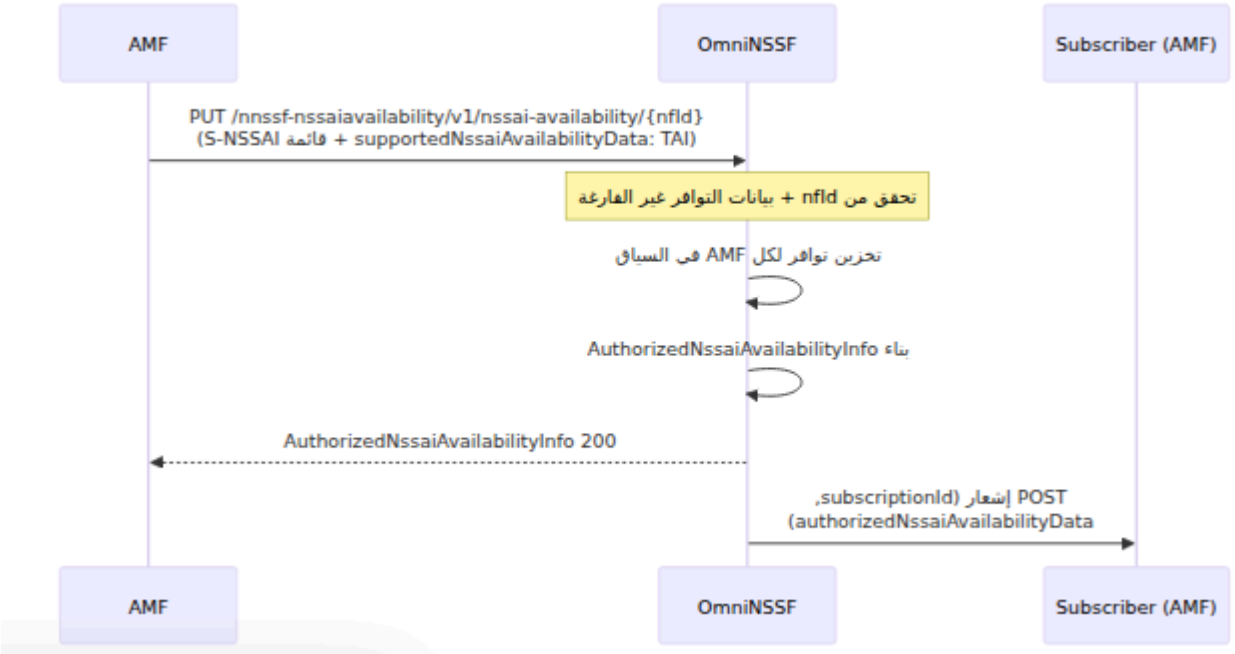
يدفع AMFs S-NSSAIs التي يدعمونها لكل TAI. تقوم NSSF بتخزين هذه المعلومات وتفويضها وتقوم بإخطار أي مشتركين بالتغيير.



تطبق PATCH عمليات المخزن التوافر على JSON Patch (RFC 6902) على السجل الحالي لكل بإعادة إشعار PATCH و PUT غير معروف `404`. يقوم كل من `{nfId}` ضد PATCH ؛ تُرجعAMF المشتركين مع بيانات التوافر المصرح بها.

NSSAI اشتراك وإخطار توافر

إشعار تغيير لكل مشترك كلما تم تحديث التوافر NSSF رد الاتصال؛ تطلق URI يسجل المشتركون المخزن بواسطة PUT أو PATCH.



لكل مشترك؛ يحمل الإشعار `nfNssaiAvailabilityUri` إلى HTTP POSTs الإشعارات هي `subscriptionId` و `authorizedNssaiAvailabilityData`. يتم تسليم كل إشعار خارج .

محاولات `nssf_notification_max_attempts` مسار الطلب مع محاولات محدودة: حتى الافتراضي `200`) `nssf_notification_backoff_ms` (الافتراضي `3`) باستخدام جدول التراجع بعد ذلك يتم إسقاط الإشعار وتسجيل الفشل. نظرًا لأن التسليم ،(مللي ثانية ثم `1000` مللي ثانية يتم خارج مسار الطلب، فإن المشترك البطيء أو غير المتاح لا يمنع تحديث التوافر الذي أدى إلى الإشعار.

تسجيل NRF ونبض القلب

معلنة عن كل من خدمات NRF، الخاص بها مع NF ملف تعريف NSSF عند بدء التشغيل، تسجل SBI، ثم تعيد التسجيل على نبض قلب دوري.



ويمكن `omni_nssf_nrf_registration_status` ي��م عكس حالة التسجيل في مقياس إجبارها على إعادة التسجيل من خلال واجهة OAM (`POST /oam/nrf/reregister`).

← نظرة عامة

استكشاف أخطاء OmniNSSF وإصلاحها

المشكلات الشائعة

العرض	السبب المحتمل	
اختيار الشريحة يعيد `400` `MANDATORY_IE_MISSING`	استعلام AMF يفتقر إلى `nf-type` أو `nf-id`	AMF تأكد من أن الإجباريين
اختيار الشريحة يعيد `400` "لا توجد معلومات شريحة مقدمة"	الاستعلام لا يحتوي على أي من `slice-info-request-for-pdu-session`، `slice-info-request-for-registration`، أو `snssai`	AM تحقق من أن للإجراء
اختيار الشريحة يعيد `403` `SNSSAI_NOT_FOUND`	S-NSSAI المطلوب ليس له إدخال NSI مطابق	أضف S-NSSAI ل `POST /o` أو عبر
الاستجابة تفتقر إلى `allowedNssaiList`	لا يوجد إدخال `allowed_nssai` لشبكة PLMN الخاصة بالاستعلام ولا يوجد تطابق احتياطي لشبكة home-PLMN	أضف PLMN (أو `allowed_nssa`
الشريحة تبدو غير متاحة في TAI	يوجد تقرير توافر AMF متعارض لذلك TAI بدون S-NSSAI	افحص `GET` `/api/nssai_av` القديمة باستخدام `/oam/nssai_av`
لا يوجد `targetAmfSet` في الاستجابة	لا يوجد إدخال `amf_set_mapping` لمفتاح S-NSSAI	أضف مفتاح `"ss` `amf_set_mapp`
`0` حالة تسجيل NRF	خاطئ `nrf_uri` أو NRF غير متاح	تحقق من `NRF_U` باستخدام `POST` `/oam/nrf/rere`
`400` التوافر PUT تم رفض	بيانات `supportedNssaiAvailabilityData` مفقودة/فارغة، أو إدخال `tai` يفتقر إلى قائمة `supportedSnssaiList` غير فارغة	شكل جسم توافر

العرض	السبب المحتمل	
PATCH `404` التوافر يعيد	تم إرسال JSON Patch لـ `nfId` بدون توافر مخزن	ف PUT قم بعمل `nfId` الصحيح
تم رفض إنشاء الاشتراك `400`	الجسم يفتقر ��لى `nfNssaiAvailabilityUri`، `taiList`، أو `event`	مطلوبة للاشتراك
PATCH/DELETE الاشتراك `404` يعيد	`subscriptionId` غير معروف	`subscr` تأكد من صلي `Location`
لم يتم إخطار المشتركين	`nfNssaiAvailabilityUri` غير متاح	شعارات مع تراجع محدود (`nssf_notifi` ت، الافتراضي `3` كانية الوصول إلى شتركين وسجلات الاستسلام