مرجع التكوين

دليل كامل لجميع معلمات التكوين

نظرة عامة على المعمارية

بوابة OmniMessage SMPP تقوم بترجمة رسائل **واجهة بروتوكول بدون حالة** هي SMPP إلى/from OmniMessage. يتم التعامل مع جميع منطق الأعمال، وقرارات التوجيه، وتخزين الرسائل بواسطة OmniMessage Core - حيث تقوم البوابة ببساطة:

1. **تستقبل** PDUs SMPP من شركات الاتصالات والعملاء
2. **ترجمها** إلى تنسيق OmniMessage عبر واجهة برمجة التطبيقات REST
3. **تستعلم** عن OmniMessage للحصول على الرسائل لإرسالها
4. **ترسل** PDUs SMPP إلى شركات الاتصالات
5. **تبلغ** OmniMessage حالة التسليم مرة أخرى إلى

هذا هو نفسه كيف تعمل واجهات OmniMessage الأخرى (Diameter، MAP، IMS) - كلها مترجمات بروتوكول بدون حالة تفوض إلى OmniMessage Core.

موقع ملف التكوين

`/opt/omnimessage-smpp/config/runtime.exs`

مهم: بعد تغيير التكوين، أعد تشغيل البوابة:

```
sudo systemctl restart omnimessage-smpp
```

هيكل التكوين

يستخدم ملف التكوين بناء جملة Elixir. الهيكل الأساسي:

```
import Config

# الإعدادات العالمية
config :omnimessage_smpp,
  setting_name: value

# عمليات الربط SMPP
config :omnimessage_smpp, :binds, [
  %{
    name: "bind_name",
    # ... إعدادات الربط
  }
]
```

الإعدادات العالمية

###API_BASE_URL

عنوان URL لمنصة OmniMessage Core

```
config :omnimessage_smpp,
  api_base_url: "https://omnimessage-core.example.com:8443"
```

المعلمة	النوع	مطلوب	الافتراضي
`api_base_url`	سلسلة (URL)	نعم	-

الغرض: عنوان URL لمنصة OmniMessage Core. تتواصل البوابة مع OmniMessage عبر واجهة برمجة التطبيقات REST لجميع معالجة الرسائل:

- **إرسال الرسائل**: إرسال رسائل OmniMessage المستلمة إلى SMPP للمعالجة
- **استرجاع الرسائل** SMPP: الاستعلام عن الرسائل الموجهة إلى شركات الاتصالات
- **الإبلاغ عن حالة التسليم**: تحديث حالة تسليم الرسالة مرة أخرى إلى OmniMessage
- **صحة النظام**: فحوصات صحية دورية

مع OmniMessage **حرج**: هنا تحصل البوابة على كل "عقلها". تتعامل:

- ✓ التحقق من صحة الرسائل وفحص التنسيق
- ✓ قرارات التوجيه (أي شركة اتصالات يجب استخدامها)
- ✓ تحديد المعدل والاختناق
- ✓ التحقق من صحة الأرقام
- ✓ تخزين الرسائل واستمراريتها
- ✓ منطق إعادة محاولة التسليم
- ✓ تتبع الحالة

تقوم البوابة ببساطة بترجمة تنسيق SMPP ↔ OmniMessage.

أمثلة:

```
# HTTPS مع IP
api_base_url: "https://192.168.1.100:8443"

# HTTPS مع اسم المضيف
api_base_url: "https://omnimessage-core.company.com:8443"

# HTTP (غير موصى به للإنتاج)
api_base_url: "http://192.168.1.100:8080"
```

متطلبات الشبكة:

- يجب أن تتمتع البوابة بإمكانية الوصول إلى شبكة OmniMessage Core
- استخدم HTTPS في الإنتاج (قم بتكوين `verify_ssl_peer`)
- يجب أن يسمح جدار الحماية بخروج HTTPS على المنفذ المحدد

منفذ إدارة/OAM API (`api_ex`)

المنفذ الخاص بواجهة API إدارة/OAM الخاصة بالبوابة

تقدم البوابة واجهة **API إدارة/OAM الخاصة بها** (`/status`, `/smpp/client_peers`, `/smpp/server_peers`, `/smpp/binds`, `/smpp/summary`) عبر `api_ex`، الافتراضي هو `8443` (`config/config.exs`). متميز عن `api_base_url` هذه **خادم** تستضيفه البوابة - الذي هو نقطة نهاية OmniMessage Core التي تستدعيها البوابة على **العميل**.

لا تخلط بين الاثنين:

الإعداد	الاتجاه	الافتراضي	المعنى
`api_base_url`	بوابة → Core (عميل)	`…:8443`	حيث ترسل البوابة/تستعلم عن الرسائل
`api_ex` `port`	بوابة (خادم)	`8443`	OAM API حيث تستمع واجهة الخاصة بالبوابة

التواجد مع OmniMessage Core: يستضيف OmniMessage Core واجهة API الخاصة بالتخزين/الرسائل على منفذ SMPP أيضًا. عندما تعمل بوابة `api_ex` `8443` على **نفس المضيف** فإن كلاهما يكون افتراضيًا على `8443` ولا يمكن أن يرتبط إلا واحد منهما - الآخر يتسبب، مثل Core، في حدوث حلقة تعطل مع `:eaddrinuse` (وتصل طلبات `/api/messages` إلى الخدمة الخاطئة، مما يعيد 404).

لإصلاح ذلك، قم بتحرير `config/runtime.exs` (المعتمد في `/etc/omnimessage_smpp/runtime.exs`) وقم بإلغاء تعليق تجاوز `api_ex`، مع تعيين منفذ مجاني حتى تنتقل واجهة OAM API الخاصة بالبوابة بعيدًا عن منفذ Core. لا يزال Core يستهدف `8443` الخاص بـ `api_base_url`:

```
# يمتلك Core :8443 لواجهة API الخاصة OAM API التخزين؛ نقل واجهة API
8444: بالبوابة إلى
config :api_ex,
  api:
    Application.get_env(:api_ex, :api, %{})
    |> Map.put(:port, 8444)
    |> Map.put(:listen_ip, "0.0.0.0")
```

إذا تم ترك التجاوز غير معلق، فإن المنفذ سيكون افتراضيًا على `8443`. لا تحتاج البوابات المستقلة (على مضيف مختلف عن Core) إلى أي تغيير.

SMPP_POLL_INTERVAL

تكرار فحص قائمة الانتظار (بالملي ثانية)

```
config :omnimessage_smpp,
  smpp_poll_interval: 100
```

المعلمة	النوع	مطلوب	الافتراضي
`smpp_poll_interval`	عدد صحيح	لا	100

الغرض: مدى تكرار (بالملي ثانية) كل عميل لفحص قائمة الرسائل.

الإرشادات:

- **حجم عالي (>100 TPS)**: 100-500ms
- **حجم متوسط (100-10 TPS)**: 500-1000ms
- **حجم منخفض (<10 TPS)**: 1000-2000ms

متغير البيئة: `SMPP_POLL_INTERVAL`

VERIFY_SSL_PEER

التحقق من شهادة SSL

```
config :omnimessage_smpp,
  verify_ssl_peer: false
```

المعلمة	النوع	مطلوب	الافتراضي
`verify_ssl_peer`	Boolean	لا	false

الغرض: ما إذا كان يجب التحقق من شهادات SSL عند الاتصال بواجهة API الخلفية.

القيم:

- `true`: (الإنتاج مع شهادات صالحة) تحقق من الشهادات
- `false`: (شهادات موقعة ذاتيًا، اختبار) تخطي التحقق

متغير البيئة: `VERIFY_SSL_PEER`

SMSC_NAME

معرف البوابة للتسجيل

```
config :omnimessage_smpp,
  smsc_name: "smpp_gateway"
```

المعلمة	النوع	مطلوب	الافتراضي
`smsc_name`	سلسلة	لا	"smpp_gateway"

الغرض: يحدد هذه النسخة من البوابة في واجهة قائمة الرسائل الخلفية.

متغير البيئة: `SMSC_NAME`

تكوين ربط عميل SMPP

(عميل) **ESME عمليات الربط الخاصة بالعميل** هي **اتصالات صادرة** حيث تعمل البوابة كـ لشركات الاتصالات. في هذا الوضع، تقوم البوابة ببدء الاتصال لإرسال **SMSC** تتصل بخوادم واستقبال الرسائل عبر شركات الاتصالات الخارجية.

مثال كامل لربط العميل

```
config :omnimessage_smpp, :binds, [
  %{
    # معرف فريد لهذه الاتصال
    name: "vodafone_uk",

    # وضع الاتصال
    mode: :client,

    # نوع ربط SMPP
    bind_type: :transceiver,

    # عنوان خادم SMPP لشركة الاتصالات
    host: "smpp.vodafone.co.uk",
    port: 2775,

    # تثبيت عنوان/منفذ المصدر المحلي للاتصال الصادر (اختياري،
    # تعيين إذا كانت شركة الاتصالات تبيض حسب IP/المنفذ المصدر)
    bind_source_ip: "10.20.30.40",
    bind_source_port: 5000,

    # بيانات الاعتماد الخاصة بالمصادقة
    system_id: "your_username",
    password: "your_password",

    # حقول بروتوكول SMPP (اختياري، تعيين إذا كانت شركة الاتصالات
تتطلب)
    system_type: "",
    addr_ton: 0,
    addr_npi: 0,
    address_range: "",

    # تحديد المعدل
    tps_limit: 100,

    # اسم قائمة الانتظار المشتركة (اختياري، الافتراضي هو اسم الربط).
تعيين نفس
    # القيمة على عدة عمليات ربط لجعلها تستنزف قائمة انتظار مشتركة
واحدة.
    queue: "vodafone_uk",

    # تكرار فحص قائمة الانتظار
```

```
    queue_check_frequency: 1000,

    # فترة إبقاء الاتصال (بالثواني، 0 لتعطيل)
    enquire_link_interval: 60,

    # تخزين الرسائل (اختياري)
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60
  }
]
```

معلمات ربط العميل

name

معرف الاتصال الفريد

النوع	مطلوب	مثال
سل��لة	نعم	`"vodafone_uk"`

الغرض: يحدد هذا الاتصال بشكل فريد SMPP.

- يستخدم في السجلات والقياسات
- يجب أن يكون فريدًا عبر جميع عمليات الربط
- استخدم أسماء وصفية (شركة الاتصالات، المنطقة، الغرض)

اتفاقيات التسمية:

- `carrier_region`: `"vodafone_uk"`, `"att_us"`
- `purpose_number`: `"marketing_1"`, `"alerts_primary"`

mode

نوع الاتصال

النوع	مطلوب	القيمة
Atom	نعم	`:client`

الغرض: يحدد هذا كاتصال صادر حيث تعمل البوابة كـ **ESME** تتصل بـ **SMSC** خارجي.

القيمة الثابتة: دائمًا `:client` للاتصالات الصادرة.

bind_type

نوع جلسة SMPP

النوع	مطلوب	القيم المسموح بها
Atom	نعم	`:transmitter`, `:receiver`, `:transceiver`

الغرض: يحدد قدرة اتجاه الرسالة.

الخيارات:

- `:transmitter` - إرسال الرسائل فقط (submit_sm)
- `:receiver` - استلام الرسائل فقط (deliver_sm)
- `:transceiver` - إرسال واستقبال (الأكثر ش��وعًا)

التوصية: استخدم `:transceiver` ما لم تتطلب شركة الاتصالات نوعًا محددًا.

host

اسم المضيف أو IP لخادم SMPP لشركة الاتصالات

النوع	مطلوب	مثال
سلسلة	نعم	`"smpp.carrier.com"` أو `"10.5.1.100"`

الغرض: عنوان خادم SMPP لشركة الاتصالات.

أمثلة:

```
host: "smpp.vodafone.co.uk"
host: "10.20.30.40"
host: "smpp-primary.carrier.net"
```

port

منفذ خادم SMPP

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	نعم	2775	1-65535

الغرض: منفذ TCP لاتصال SMPP.

المنفذ القياسي: 2775

أمثلة:

```
port: 2775  # قياسي
port: 3000  # مخصص
```

bind_source_ip

المصدر المحلي للاتصال الصادر IP

النوع	مطلوب	الافتراضي	مثال
سلسلة	لا	(افتراضي النظام)	`"10.20.30.40"`

الغرض: يثبت عنوان IP المصدر المحلي المستخدم للاتصال TCP الصادر إلى شركة الاتصالات. قم ب��عيين هذا عندما تبيض شركة الاتصالات الاتصالات حسب IP المصدر ولدى مضيف البوابة عدة عناوين. يتم تسجيل العنوان غير الصالح وتجاهله (المصدر IP يختار النظام).

ملاحظة: هذا حقل خاص بالتكوين فقط (يتم قراءته من تكوين تطبيق `:binds`); لا يمكن تحريره من واجهة الويب / تخزين Mnesia.

bind_source_port

المنفذ المصدر المحلي للاتصال الصادر

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	(عابر)	1-65535

الغرض: يثبت المنفذ المصدر المحلي المستخدم للاتصال الصادر. قم بتعيين هذا عندما تبيض TCP المصدر **و** المنفذ. يتم فتح المقبس مع IP شركة الاتصالات حسب بحيث يمكن `SO_REUSEADDR` بينما `:eaddrinuse` بدلاً من الفشل مع) إعادة ربط المنفذ المثبت على الفور عند إعادة الاتصال يكون المقبس السابق في `TIME_WAIT`).

ملاحظة: مثل `bind_source_ip`، هذا حقل خاص بالتكوين فقط ولا يمكن تحريره من واجهة الويب.

system_id

اسم المستخدم للمصادقة

النوع	مطلوب	مثال
سلسلة	نعم	`"company_user"`

الغرض: اسم المستخدم المقدم من شركة الاتصالات للمصادقة.

الأمان: احمِ هذه البيانات - مخزنة في ملف التكوين.

password

كلمة مرور المصادقة

النوع	مطلوب	مثال
سلسلة	نعم	`"secret_password"`

الغرض: كلمة المرور المقدمة من شركة الاتصالات للمصادقة.

الأمان:

- احمِ هذه البيانات
- استخدم كلمات مرور قوية
- قم بتدويرها بشكل دوري

tps_limit

حدود المعاملات في الثانية

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	نعم	100	1-10000

الغرض: الحد الأقصى لعدد الرسائل في الثانية التي يمكن إرسالها عبر هذا الاتصال.

الإرشادات:

- قم بتعيينها إلى 70-80% من الحد الأقصى لشركة الاتصالات
- يمنع الاختناق/قطع الاتص��ل
- يوفر مساحة للتسليم الإيصالات

أمثلة:

```
tps_limit: 10    # حجم منخفض
tps_limit: 50    # حجم متوسط
tps_limit: 100   # (الأكثر شيوعًا) حجم عالي
tps_limit: 1000  # حجم عالي جدًا
```

الحساب:

```
100 = إذا كان الحد الأقصى لشركة الاتصالات TPS
قم بتعيين tps_limit = 70-80
كمساحة TPS يترك 30-20
```

queue

اسم قائمة الانتظار المشتركة

النوع	مطلوب	الافتراضي	مثال
سلسلة	لا	اسم الربط	`"sinch.smpp"`

الغرض: قائمة الانتظار التي تستنزف منها هذه الربط من الخلفية. الافتراضي هو الخاص `name` على عدة ربط `queue` بالربط، لذا تستنزف كل ربط قائمة انتظارها الخاصة. قم بتعيين نفس قيمة Sinch لجعلها جميعًا تستنزف قائمة انتظار مشتركة واحدة - على سبيل المثال، اثنان من اتصالات لمشاركة الحمل/التكرار `"sinch.smpp"` تستنزف كلاهما.

أمثلة:

```
# ربطان يشتركان في قائمة انتظار واحدة
%{name: "sinch_1", queue: "sinch.smpp", ...}
%{name: "sinch_2", queue: "sinch.smpp", ...}
```

queue_check_frequency

فترة استعلام قائمة انتظار الرسائل (بالملي ثانية)

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	نعم	1000	100-10000

الغرض: مدى تكرار التحقق من الخلفية للحصول على رسائل جديدة لإرسالها.

الإرشادات:

- **حجم عالي (>100 TPS)**: 500-1000ms
- **حجم متوسط (100-10 TPS)**: 1000-2000ms
- **حجم منخفض (<10 TPS)**: 2000-5000ms

المقايضات:

- قيمة أقل = أسرع التقاط الرسائل، مزيد من الحمل على واجهة برمجة التطبيقات
- قيمة أعلى = أبطأ التقاط، أقل حمل على واجهة برمجة التطبيقات

enquire_link_interval

فترة إبقاء SMPP (بالثواني)

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	60	0-3600

الغرض: مدى تكرار (بالثواني) إرسال SMPP enquire_link PDUs للتحقق من أن الاتصال نشط. يستجيب الخادم البعيد بـ enquire_link_resp.

لإرشادات:

- **الافتراضي (60)**: مناسب لمعظم شركات الاتصالات
- **قيم أقل (15-30)**: كشف أسرع عن الفشل، مزيد من الحركة
- **قيم أعلى (120-300)**: أقل حمل، كشف أبطأ عن الفشل
- **0**: يعطل enquire_link (غير موصى به) تمامًا

أمثلة:

```
enquire_link_interval: 60   # (1 دقيقة) قياسي
enquire_link_interval: 30   # إبقاء نشط عدواني
enquire_link_interval: 0    # معطل
```

system_type

معرف نوع نظام SMPP

النوع	مطلوب	الافتراضي	مثال
سلسلة	لا	`""`	`"0TP"`

الغرض: حقل بروتوكول SMPP المرسل أثناء الربط. تتطلب بعض شركات الاتصالات قيمة محددة. اتركه فارغًا ما لم تحدد شركة الاتصالات واحدة.

addr_ton

نوع عنوان الرقم

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	0	0-6

الغرض: حقل بروتوكول SMPP يحدد نوع الرقم المستخدم في طلب الربط.

القيم الشائعة:

- `0` - غير معروف
- `1` - دولي
- `2` - وطني
- `5` - أبجدي رقمي

قم بتعيينه كما تتطلب شركة الاتصالات.

addr_npi

مؤشر خطة ترقيم العنوان

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	0	0-18

الغرض: حقل بروتوكول SMPP يحدد خطة الترقيم في طلب الربط.

القيم الشائعة:

- `0` - غير معروف
- `1` - ISDN/E.164
- `3` - بيانات/X.121
- `9` - خاص

قم بتعيينه كما تتطلب شركة الاتصالات.

address_range

نطاق العنوان للربط

النوع	مطلوب	الافتراضي	مثال
سلسلة	لا	`""`	`"614*"`

الغرض: حقل بروتوكول SMPP يحدد نطاق العنوان الذي تتعامل معه هذه الربط. تستخدمه بعض شركات الاتصالات لتصفية الرسائل التي يتم تسليمها إلى هذا الاتصال. اتركه فارغًا ما لم تحدد شركة الاتصالات قيمة.

enabled

حالة تمكين النظير

النوع	مطلوب	الافتراضي
Boolean	لا	`true`

الغرض: يتحكم فيما إذا كان هذا النظير نشطًا. يتم الاحتفاظ بالنظائر المعطلة في التكوين ولكن لا تقيم اتصالات. مفيد لأخذ اتصال مؤقتًا دون حذف تكوينه.

cache_enabled

تمكين التخزين المؤقت للرسائل المحلية

النوع	مطلوب	الافتراضي
Boolean	لا	`false`

الغرض: عند التمكين، يتم تخزين الرسائل الواردة مؤقتًا محليًا إذا كانت واجهة API الخلفية غير متاحة، ثم يتم تسليمها تلقائيًا عند استعادة الاتصال. انظر MESSAGE_CACHE.md للحصول على التفاصيل الكاملة.

cache_max_size

الحد الأقصى للرسائل المخزنة مؤقتًا

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	10000	1-1000000

الغرض: الحد الأقصى لعدد الرسائل التي يمكن تخزينها مؤقتًا لكل ربط. عند الوصول إلى الحد، يتم طرد أقدم الرسائل (FIFO). ينطبق فقط عندما يكون `cache_enabled` هو `true`.

cache_retry_interval

فترة إعادة المحاولة الأساسية (بالثواني)

النوع	مطلوب	الافتراضي
عدد صحيح	لا	60

الغرض: الفترة الأساسية بالثواني قبل إعادة محاولة تسليم رسالة مخزنة مؤقتًا. يتم دمجها مع التراجع الأسي (إعادة المحاولة 0: 60 ثانية، إعادة المحاولة 1: 120 ثانية، إعادة المحاولة 2: 240 ثانية، إلخ). ينطبق فقط عندما يكون `cache_enabled` هو `true`.

مثال واجهة الويب:

تكوين ربط خادم SMPP

عمليات الربط الخاصة بالخادم SMSC (خادم) تحدد **الاتصالات الواردة** حيث تعمل البوابة كـ خارجيين. في هذا الوضع، تتصل الأنظمة الشريكة بالبوابة (عملاء) **ESMEs** تقبل الاتصالات من لإرسال واستقبال الرسائل.

مثال كامل لربط الخادم

```
config :omnimessage_smpp, :server_binds, [
  %{
    # معرف ��ريد لهذا العميل
    name: "partner_acme",

    # بيانات الاعتماد المتوقعة من العميل
    system_id: "acme_corp",
    password: "acme_secret",

    # أنواع الربط المسموح بها
    allowed_bind_types: [:transmitter, :receiver, :transceiver],

    # قيود IP
    ip_whitelist: ["192.168.1.0/24", "10.50.1.100"],

    # قيود عنوان المصدر (فارغ = السماح للجميع)
    source_address_whitelist: [],

    # تحديد المعدل
    tps_limit: 50,

    # تكرار فحص قائمة الانتظار
    queue_check_frequency: 1000,

    # فترة إبقاء الاتصال (بالثواني، 0 لتعطيل)
    enquire_link_interval: 60,

    # تخزين الرسائل (اختياري)
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60,

    # إيصالات التسليم (DLR) (اختياري) مرة أخرى إلى هذا الربط
    deliver_receipts: false
  }
]
```

معلمات ربط الخادم

name

معرف العميل

النوع	مطلوب	مثال
سلسلة	نعم	`"partner_acme"`

الغرض: يحدد العميل الخارجي الذي يتصل بك.

اتفاقيات التسمية: استخدم اسم الشريك/العميل لتسهيل التعرف.

system_id

اسم المستخدم المتوقع من العميل

النوع	مطلوب	مثال
سلسلة	نعم	`"acme_corp"`

الغرض: اسم المستخدم الذي يجب أن يقدمه العميل الخارجي للمصادقة.

توفير للعميل: شارك هذه البيانات مع شريكك.

password

كلمة المرور المتوقعة من العميل

النوع	مطلوب	مثال
سلسلة	نعم	`"secure_password"`

الغرض: كلمة المرور التي يجب أن يقدمها العميل الخارجي للمصادقة.

الأمان:

- استخدم كلمات مرور قوية
- فريدة لكل عميل
- شاركها بشكل آمن مع الشريك

allowed_bind_types

أنواع الجلسات المسموح بها

النوع	مطلوب	الافتراضي
قائمة من الذرات	نعم	-

الغرض: يقيد أنواع الربط التي يمكن للعميل استخدامها.

الخيارات:

```
allowed_bind_types: [:transceiver]  # فقط transceiver
allowed_bind_types: [:transmitter, :receiver]  # TX أو RX
allowed_bind_types: [:transmitter, :receiver, :transceiver]  # أي
```

التوصية: السماح بالثلاثة ما لم تحتاج إلى قيود.

ip_whitelist

عناوين IP المسموح بها للعميل

النوع	مطلوب	الافتراضي	التنسيق
قائمة من السلاسل	نعم	`[]`	CIDR أو صيغة IPs

الغرض: الأمان - السماح فقط بالاتصالات من IPs المعروفة.

التنسيقات:

- فردي: `"192.168.1.100"` (تلقائيًا `/32`) IP
- شبكة CIDR: `"192.168.1.0/24"`، `"10.0.0.0/8"`
- مزيج من الاثنين: `["10.50.1.100" ,"192.168.1.0/24"]`

أمثلة:

```
# السماح بأي IP (غير موصى به)
ip_whitelist: []

# IP فردي
ip_whitelist: ["203.0.113.50"]

# عدة IPs
ip_whitelist: ["203.0.113.50", "203.0.113.51"]

# شبكة
ip_whitelist: ["192.168.1.0/24"]

# مختلط
ip_whitelist: ["192.168.1.0/24", "10.50.1.100", "10.60.0.0/16"]
```

الشبكات الشائعة:

- `/32` - IP فردي (بدون قناع IP تلقائي لعناوين)
- `/24` - 256 عنوانًا (مثل 192.168.1.0-255)
- `/16` - 65,536 عنوانًا (مثل 10.50.0.0-255.255)
- `/8` - 16,777,216 عنوانًا (مثل 10.0.0.0-255.255.255.255)

source_address_whitelist

العناوين المسموح بها

التنسيق	الافتراضي	مطلوب	النوع
أنماط Regex	`[]`	لا	قائمة من السلاسل

الغرض: يحدد العناوين المصدرية (معرفات المرسل) التي يمكن للعملاء المتصلين استخدامها عند إرسال الرسائل. القائمة الفارغة تسمح بجميع العناوين. كل إدخال هو نمط تعبير منتظم.

أمثلة:

```
# السماح بأي عنوان مصدر
source_address_whitelist: []

# المطابقة الدقيقة (مربوطة)
source_address_whitelist: ["^MyBrand$", "^AlertService$"]

# المطابقة بالبادئة
source_address_whitelist: ["^614", "^\+61"]

# نطاق الأرقام مع فئة الأحرف
source_address_whitelist: ["^61[45]\d{8}$"]

# مختلط
source_address_whitelist: ["^MyBrand$", "^614", "^\+61400000001$"]
```

تُرفض الرسائل التي تحتوي على عناوين مصدر غير مسموح بها مع `ESME_RINVSRCADR`. راجع SOURCE_ADDRESS_WHITELIST.md للحصول على التفاصيل الكاملة.

tps_limit

حد الرسائل في الثانية

نفس إعداد `tps_limit` لربط العميل - يتحكم في معدل تسليم `deliver_sm` للعملاء المتصلين.

queue_check_frequency

فترة استعلام قائمة الانتظار

نفس إعداد `queue_check_frequency` لربط العميل - مدى تكرار التحقق من الرسائل لتسليمها إلى هذا العميل.

enquire_link_interval

فترة إبقاء الاتصال (بالثواني)

نفس إعداد `enquire_link_interval` لربط العميل. يتحكم في مدى تكرار إرسال الخادم لـ enquire_link PDUs للعملاء المتصلين للتحقق من أنهم لا يزالون على قيد الحياة.

enabled

حالة تمكين النظير

نفس إعداد `enabled` لربط العميل. لا تقبل نظائر الخادم المعطلة الاتصالات الواردة.

cache_enabled

تمكين التخزين المؤقت المحلي للرسائل

نفس إعداد `cache_enabled` لربط العميل. راجع MESSAGE_CACHE.md.

cache_max_size

أقصى عدد من الرسائل المخزنة مؤقتًا

نفس إعداد `cache_max_size` لربط العميل.

cache_retry_interval

فترة إعادة المحاولة الأساسية (بالثواني)

نفس إعداد `cache_retry_interval` لربط العميل.

deliver_receipts

إرسال إيصالات تسليم SMPP (DLRs) إلى هذا الربط

النوع	مطلوب	الافتراضي
بولياني	لا	`false`

الغرض: يرسل البوابة إيصال تسليم SMPP مرة أخرى إلى هذا الربط لكل رسالة يقوم بإرسالها. تتعقب البوابة كل رسالة تم إرسالها وترسل الإيصال عندما تصل الرسالة إلى حالة نهائية. الإيصال هو PDU `deliver_sm` مع تعيين `esm_class` إلى `0x04` (إيصال تسليم SMSC)، وفقًا لـ SMPP v3.4 القسم 5.2.12.

متى تستخدم: قم بتعيين `deliver_receipts: true` لربط الاتصال المخصص الذي يحتاج للربط الذي لا يحت��ج مشغله إلى تأكيد التسليم لكل رسالة. احتفظ بالقيمة الافتراضية `false` إلى إيصالات.

المتطلبات: يجب أن يعمل الربط مع `cache_enabled: false`. يستخدم الإيصال معرف الرسالة الذي أعادته البوابة في `submit_sm_resp`. تعيد الذاكرة المؤقتة معرفًا مؤقتًا لا يتطابق مع معرف النواة، لذا فإن الربط المخزن مؤقتًا يكسر ارتباط الإيصالات. إذا قام الربط بتعيين

تكتب البوابة تحذيرًا عند بدء، `cache_enabled: true` و `deliver_receipts: true` التشغيل ولا يتطابق معرف الإيصال.

كيف يعمل: يرسل الربط المرسل `submit_sm`. تقبل البوابة الرسالة وتعيد `submit_sm_resp` مع معرف الرسالة. تتعقب البوابة الرسالة وتستعلم عن حالة OmniMessage core. عندما يتم تسليم الرسالة أو انتهاء صلاحيتها أو رفضها، ترسل البوابة إيصال `deliver_sm` مرة أخرى إلى الربط. يحمل الإيصال نفس معرف الرسالة وحقل الحالة (`stat:DELIVRD`، `stat:EXPIRED`، `stat:UNDELIV`، أو `stat:REJECTD`).

يحتفظ المتعقب بالحالة في الذاكرة فقط. إذا أعادت البوابة التشغيل، تُفقد الإيصالات للرسائل التي كانت في الهواء في تلك اللحظة. هذه آلية تعتمد على الجهد الأفضل.

تعيين الحالة إلى الإيصال:

حالة النواة	إيصال `stat`
`delivered`	`DELIVRD`
`expired`	`EXPIRED`
`dropped`	`UNDELIV`
`rejected`	`REJECTD`
`balance_rejected`	`REJECTD`

راجع USAGE.md و MONITORING.md للحصول على تدفق الرسائل من البداية إلى النهاية و المقاييس ذات الصلة.

مثال واجهة الويب:

تكوين استماع الخادم

عندما يتم تكوين روابط الخادم، تستمع البوابة للاتصالات الواردة.

مثال استماع كامل

```
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

معلمات الاستماع

host

عنوان IP للربط

النوع	مطلوب	الافتراضي	القيم الشائعة
سلسلة	لا	`"0.0.0.0"`	`"0.0.0.0"`, `"127.0.0.1"`

الغرض: أي واجهة شبكة للاستماع عليها.

القيم:

- `"0.0.0.0"` - الاستماع على جميع الواجهات (موصى به)
- `"127.0.0.1"` - الاستماع على localhost فقط (اختبار)
- `"192.168.1.10"` - الاستماع على IP محدد

port

منفذ TCP للاستماع عليه

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	2775	1-65535

الغرض: المنفذ للاتصالات الواردة SMPP.

المعيار: 2775

max_connections

أقصى عدد من الاتصالات المتزامنة

النوع	مطلوب	الافتراضي	النطاق
عدد صحيح	لا	100	1-10000

الغرض: يحدد العدد الإجمالي للاتصالات المتزامنة للعملاء.

الإرشادات:

- تعيين بناءً على العملاء المتوقعين
- القيم الأعلى تستخدم المزيد من الذاكرة
- المعتاد: 10-100 اتصال

أمثلة تكوين كاملة

المثال 1: اتصال موفر و��حد

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443",
  verify_ssl_peer: true,
  smsc_name: "smpp_prod"

config :omnimessage_smpp, :binds, [
  %{
    name: "att_primary",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "company_user",
    password: "secure_pass_123",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]
```

المثال 2: موفرو خدمات متعددون

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443"

config :omnimessage_smpp, :binds, [
  # أمريكا الشمالية
  %{
    name: "att_us",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "att_username",
    password: "att_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  },

  # أوروبا
  %{
    name: "vodafone_uk",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.vodafone.co.uk",
    port: 2775,
    system_id: "voda_username",
    password: "voda_password",
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]
```

المثال 3: بوابة مع روابط خادم

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443"

# الاتصالات الصادرة
config :omnimessage_smpp, :binds, [
  %{
    name: "upstream_carrier",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.carrier.com",
    port: 2775,
    system_id: "my_username",
    password: "my_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]

# تعريفات العملاء الواردة
config :omnimessage_smpp, :server_binds, [
  %{
    name: "partner_alpha",
    system_id: "alpha_corp",
    password: "alpha_secret",
    allowed_bind_types: [:transmitter, :receiver, :transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  },
  %{
    name: "partner_beta",
    system_id: "beta_inc",
    password: "beta_password",
    allowed_bind_types: [:transceiver],
    ip_whitelist: ["198.51.100.50"],
    tps_limit: 25,
    queue_check_frequency: 2000
  }
]
```

```
# استماع الخادم
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

المثال 4: قائمة انتظار مشتركة مع ربط المصدر

لتقاس�� (`sinch.smpp`) اتصالان بنفس الموفر، كلاهما يستنزف قائمة انتظار مشتركة واحدة المصدر **و** المنفذ، لذا يقوم كل ربط IP الحمل والموثوقية. يقوم الموفر بإدراج العناوين بناءً على بتثبيت منفذ مصدر محلي متميز على نفس IP المصدر.

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443"

config :omnimessage_smpp, :binds, [
  %{
    name: "sinch_1",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.sinch.com",
    port: 2775,
    system_id: "sinch_user",
    password: "sinch_pass",
    tps_limit: 100,
    queue_check_frequency: 1000,

    # كلا الربطين يستنزفان نفس قائمة الانتظار الخلفية
    queue: "sinch.smpp",

    # (الموفر يدرج بناءً على كليهما) عنوان/منفذ المصدر المثبت
    bind_source_ip: "10.20.30.40",
    bind_source_port: 5001
  },
  %{
    name: "sinch_2",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.sinch.com",
    port: 2775,
    system_id: "sinch_user",
    password: "sinch_pass",
    tps_limit: 100,
    queue_check_frequency: 1000,

    # نفس قائمة الانتظار المشتركة مثل sinch_1
    queue: "sinch.smpp",

    # نفس IP المصدر، منفذ مصدر متميز
    bind_source_ip: "10.20.30.40",
    bind_source_port: 5002
```

```
  }
]
```

مفتاح التوجيه (`sinch.smpp`) **كيف يعمل**: يقوم كلا الربطين بتسجيل واجهة أمامية واحدة تُسمى `dest_smsc: "sinch.smpp"` (المشترك الخاص بهم). يتم استنزاف الرسائل التي تحمل بواسطة الربط الذي يلتقطها أولاً، مما يوزع الحمل عبر الاتصالين. إذا انقطع أحد الاتصالات، يستمر الآخر في استنزاف قائمة الانتظار. يقدم كل اتصال عنوان IP ثابت ومنفذًا للمزود للإدراج؛ يسمح `SO_REUSEADDR` بإعادة ربط منفذ مثبت على الفور عند إعادة الاتصال.

راجع `queue`، `bind_source_ip`، و`bind_source_port` للحصول على تفاصيل المعلمات.

التحقق من التكوين

بعد تحرير التكوين، تحقق قبل إعادة التشغيل:

تحقق من الصيغة

```
# تحقق من صيغة Elixir
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

إذا كانت الصيغة غير صحيحة، سترى خطأ. قم بإصلاحه قبل إعادة التشغيل.

اختبار التكوين

```
# إعادة التشغيل في الواجهة الأمامية لرؤية الأخطاء
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-
smpp console
```

اضغط على `Ctrl+C` مرتين للخروج.

متغيرات البيئة

يمكن تجاوز جميع الإعدادات العالمية باستخدام متغيرات البيئة. قم بتعيين هذه في ملف وحدة systemd الخاص بك أو بيئة shell قبل بدء البوابة.

متغير البيئة	مفتاح التكوين	راضي
`API_BASE_URL`	`api_base_url`	`"https://10.17`
`SMSC_NAME`	`smsc_name`	`"smpp_gateway"`
`SMPP_POLL_INTERVAL`	`smpp_poll_interval`	`100`
`VERIFY_SSL_PEER`	`verify_ssl_peer`	`false`
`CACHE_FLUSH_INTERVAL`	`cache_flush_interval`	`10000`
`CACHE_MAX_RETRY_ATTEMPTS`	`cache_max_retry_attempts`	`10`
`CACHE_BACKOFF_MULTIPLIER`	`cache_backoff_multiplier`	`2`

متغير البيئة	مفتاح التكوين	راضي
`MNESIA_STORAGE_TYPE`	`mnesia_storage_type`	`disc_copies`

مثال على تجاوز systemd:

```
sudo systemctl edit omnimessage-smpp
```

```
[Service]
Environment="API_BASE_URL=https://omnimessage-
core.company.com:8443"
Environment="SMSC_NAME=smpp_prod_01"
Environment="VERIFY_SSL_PEER=true"
```

أفضل الممارسات الأمنية

1. **حماية ملف التكوين**:

   ```
   sudo chmod 600 /opt/omnimessage-smpp/config/runtime.exs
   sudo chown omnimessage-smpp:omnimessage-smpp
   /opt/omnimessage-smpp/config/runtime.exs
   ```

2. **استخدام كلمات مرور قوية**:
   - الحد الأدنى 12 حرفًا
   - مزج الحروف والأرقام والرموز
   - فريدة لكل اتصال
3. **استخدام قوائم بيضاء IP**:

- قم دائمًا بتكوين `ip_whitelist` للربط بالخادم
- لا تستخدم قائمة فارغة `[]` في الإنتاج

4. **تمكين التحقق من SSL**:
   - قم بتعيين `verify_ssl_peer: true` مع شهادات صالحة
5. **تدوير الاعتماديات بانتظام**:
   - تغيير كلمات المرور كل ثلاثة أشهر
   - التنسيق مع الموفرين/الشركاء

الخطوات التالية

- مراجعة MONITORING.md لتكوين المقاييس
- قراءة USAGE.md لإدارة الاتصالات
- الاطلاع على TROUBLESHOOTING.md للمشاكل الشائعة
- العودة إلى README.md للحصول على نظرة عامة

مسرد

المصطلحات والتعريفات

A

API (واجهة برمجة التطبيقات) واجهة تُستخدم للتواصل مع نظام خلفية قائمة الرسائل.

Auto-Scroll ميزة في واجهة المستخدم على الويب في علامة التبويب السجلات التي تقوم بالتمرير تلقائيًا لعرض أحدث إدخالات السجل.

B

Backend نظام قائمة الرسائل الذي يتصل به بوابة SMPP لاسترجاع الرسائل وتخزينها.

Bind اتصال SMPP واستقبال بين نظامين. يمكن أن يكون مرسلًا، مستلمًا، أو جهاز إرسال واستقبال.

Bind Type نوع جلسة SMPP:

- **Transmitter**: إرسال الرسائل فقط
- **Receiver**: استلام الرسائل فقط
- **Transceiver**: إرسال واستقبال الرسائل

Bind Failure عندما تفشل محاولة المصادقة على SMPP، عادةً بسبب بي��نات اعتماد غير صحيحة أو قيود على عنوان IP.

C

CIDR (توجيه بين المجالات بدون فئات) ترميز لتحديد نطاقات عناوين IP (على سبيل المثال، `192.168.1.0/24` يمثل 256 عنوان IP).

Client Bind اتصال SMPP صادر حيث تعمل البوابة كـ **ESME** تتصل بـ **SMSC** خارجي (عادةً خادم SMPP لمشغل الاتصالات). في هذا الوضع، تكون البوابة هي العميل.

Connection Status الحالة الحالية لارتباط SMPP:

- **Connected**: نشط وعامل
- **Disconnected**: غير متصل
- **Reconnecting**: محاولة لإعادة إنشاء الاتصال

Counter مقياس يزيد فقط (يتم إعادة تعيينه عند إعادة تشغيل الخدمة)، يُستخدم للإجماليات مثل الرسائل المرسلة.

D

Data Coding حقل SMPP يحدد ترميز أحرف الرسالة (GSM-7، UCS-2، إلخ).

Deliver_SM PDU SMPP يُرسل بواسطة SMSC (الخادم) لتسليم رسالة إلى ESME متصل (العميل). يُستخدم بواسطة ارتباطات الخادم لدفع الرسائل إلى الشركاء المتصلين.

Delivery Failure عندما لا يمكن تسل��م رسالة، مما يشير إليه رد خطأ من المشغل.

Delivery Receipt (DLR) تأكيد من المشغل حول حالة تسليم الرسالة.

deliver_time حقل في قائمة الرسائل يحمل الطابع الزمني الذي تم فيه تسليم الرسالة. يتم تعيينه بواسطة OmniMessage Core عندما تبلغ البوابة عن تسليم ناجح. هذا هو اسم الحقل الذي يتوقعه النظام الخلفي عند وضع علامة على الرسالة كتم تسليمها.

dest_smsc حقل في قائمة الرسائل يشير إلى أي اتصال SMPP. يجب أن يتعامل مع الرسالة. يجب أن تساوي قيمته **Routing Key** الذي تم تسجيل النظير المستهدف تحته (انظر Frontend Registration).

Disconnection عندما يتم إنهاء اتصال SMPP نشط، سواء عن عمد أو بسبب خطأ.

E

Enquire Link رسالة SMPP تُرسل بشكل دوري للتحقق من أن الاتصال للحفاظ على الاتصال نشط.

ESM Class حقل SMPP يشير إلى نوع الرسالة وميزاتها.

ESME (كيان الرسائل القصيرة الخارجي) في مصطلحات SMPP، العميل الذي يتصل بـ SMSC لإرسال أو استلام الرسائل. عندما تعمل البوابة في **وضع العميل**، تعمل كـ ESME التطبي��

تتصل بـ SMSCs للمشغلين. عندما تعمل في **وضع الخادم**، تقبل الاتصالات من ESMEs الخارجية.

Exponential Backoff استراتيجية إعادة المحاولة حيث يتضاعف وقت الانتظار بعد كل فشل (1 دقيقة، 2 دقيقة، 4 دقائق، 8 دقائق...).

F

Firewall نظام أمان الشبكة الذي يتحكم في حركة المرور الواردة والصادرة.

Frontend Registration العملية التي تسجل بها بوابة SMPP نفسها مع OmniMessage Core. يتم إرسال نبضة كل 60 ثانية للحفاظ على التسجيل نشطًا. إذا توقفت البوابة، تنتهي صلاحية التسجيل بعد 90 ثانية ويتوقف OmniMessage عن توجيه الرسائل إليها. يتم تسجيل كل نظير تحت **Routing Key** الخاصة به بحيث يساوي الاسم المسجل `dest_smsc` الذي يتجه إليه النظام الخلفي: نظائر العملاء تحت `queue` الخاصة بهم (أو اسم النظير العاري)، وارتباطات الخادم تحت اسم الربط الخاص بهم.

G

Gateway تطبيق بوابة SMPP الذي يربط بين قائمة الرسائل والشبكات المحمولة.

Gauge مقياس يمكن أن يزيد أو ينقص، يمثل القيمة الحالية (على سبيل المثال، حالة الاتصال).

Grafana أداة تصور شائعة لعرض مقاييس Prometheus في لوحات المعلومات.

GSM-7 ترميز أحرف قياسي 7 بت للرسائل القصيرة، يدعم حتى 160 حرفًا لكل رسالة.

H

HTTP/HTTPS البروتوكولات المستخدمة للتواصل عبر الويب. HTTPS هو النسخة المشفرة.

I

IP Whitelist قائمة IP (ميزة أمان) بعناوين المسموح بها التي يمكنها الاتصال بالبوابة.

ISDN (شبكة الخدمات الرقمية المتكاملة) خطة ترقيم تُستخدم عادةً لأرقام الهواتف.

J

(لا توجد مصطلحات)

K

Keepalive رسائل دورية (enquire_link) تُرسل للحفاظ على الاتصال واكتشاف الأخطاء.

KPI (مؤشر الأداء الرئيسي) قيمة قابلة للقياس تشير إلى أداء النظام (على سبيل المثال، معدل نجاح التسليم).

L

Label في Prometheus، أزواج المفتاح والقيمة المرفقة بالمقاييس للتعريف (على سبيل المثال، `bind_name="vodafone_uk"`).

LiveView تكنولوجيا إطار Phoenix المستخدمة لتحديث واجهة المستخدم على الويب في الوقت الحقيقي.

M

Message Queue نظام خلفية يخزن الرسائل التي تنتظر الإرسال أو الاستلام.

Metrics قياسات كمية لأداء النظام، تُعرض بتنسيق Prometheus.

MO (الرسائل المنشأة من الهاتف المحمول) الرسائل المرسلة من الهواتف المحمولة إلى البوابة (واردة).

MT (الرسائل الموجهة إلى الهاتف المحمول) الرسائل المرسلة من البوابة إلى الهواتف المحمولة (صادرة).

MSISDN (رقم الدليل الدولي للمشترك في المحطة المحمولة) تنسيق قياسي لأرقام الهواتف المحمولة.

N

NPI (مؤشر خطة الترقيم) حقل SMPP يحدد نظام الترقيم (على سبيل المثال، ISDN).

O

Outbound الرسائل المتدفقة ��ن البوابة إلى الشبكات المحمولة.

Inbound الرسائل المتدفقة من الشبكات المحمولة إلى البوابة.

P

PDU (وحدة بيانات البروتوكول) حزمة رسالة SMPP فردية (على سبيل المثال، submit_sm، deliver_sm).

Prometheus نظام مراقبة مفتوح المصدر يجمع ويخزن مقاييس السلاسل الزمنية.

Q

Queue قائمة الرسائل التي تنتظر المعالجة أو الإرسال.

Queue Check Frequency مدى تكرار (بالملي ثانية) استعلام البوابة للنظام الخلفي عن رسائل جديدة.

Queue Worker مكون يسترجع الرسائل من القائمة ويرسلها عبر SMPP.

R

Rate Limiting التحكم في معدل مرور الرسائل للامتثال لقيود المشغل. انظر TPS.

Receiver نوع ارتباط SMPP الذي يستقبل الرسائل فقط (deliver_sm).

Reconnect إعادة إنشاء اتصال SMPP مفصول.

Retry محاولة إرسال رسالة فاشلة مرة أخرى، عادةً مع زيادة زمن الانتظار بشكل أسي.

Routing Key مع ��أمامي الاسم الذي يسجله النظير كواجهة OmniMessage Core، التي توجه الرسائل إليه. بالنسبة لنظير العميل، يكون هذا هو `queue` وبالتالي قيمة `dest_smsc` بالنسبة لارتباط الخادم، يكون هو ;(`queue` أو اسم النظير العاري عندما لا يتم تعيين) المكونة له اسم الربط.

S

Sequence Number معرف رقمي فريد يُخصص لكل PDU SMPP ضمن جلسة. يُستخدم على سبيل المثال، مطابقة) لمطابقة الطلبات مع ردودها submit_sm مع submit_sm_resp).

Server Bind تكوين يسمح لـ **ESMEs** الخارجية (العملاء) بالاتصال بالبوابة. في هذا الوضع، تعمل البوابة كـ **SMSC** (خادم) تقبل الاتصالات الواردة من الأنظمة الشريكة.

Session اتصال SMPP نشط بين نظامين.

Shared Queue قائمة خلفية واحدة يتم تفريغها بواسطة أكثر من ارتباط عميل واحد. تختار الارتباطات الانضمام عن طريق تعيين نفس قيمة `queue` في تكوينها على سبيل المثال، اتصالان) مما يسمح بمشاركة الحمل والازدواج��ة عبر ،(`"sinch.smpp"` لمشغلين كلاهما يقوم بتفريغ الاتصالات. يتم تعيين القيمة الافتراضية إلى اسم الربط الخاص بها، وفي هذه الحالة، يكون لكل ربط قائمته الخاصة.

Short Code عنوان SMS مختصر (عادةً 5-6 أرقام) يُستخدم للرسائل من التطبيق إلى الشخص. **غير معروف** حتى لا يسيء TON المكونة من خمسة أرقام مع `deliver_sm` يتم إرسال وجهات المشغلون تفسيرها كأرقام دولية.

Source Binding لاتصال صادر (عميل) عبر معلمات TCP المحلي وميناء IP تثبيت عنوان `bind_source_ip` و `bind_source_port`. يُستخدم عندما يقوم المشغل بإدراج الاتصالات حسب عنوان IP والميناء المصدر. انظر CONFIGURATION.md.

source_smsc حقل في قائمة الرسائل يشير إلى أي ارتباط خادم يجب أن يسلم الرسالة إلى عملائه المتصلين عبر deliver_sm.

SMPP (الرسائل القصيرة من نظير إلى نظير) بروتوكول قياسي في الصناعة لتبادل الرسائل القصيرة بين الأنظمة.

SMSC (مركز خدمة الرسائل القصيرة) في مصطلحات SMPP، المكون الخادم الذي يقبل ويتعامل مع توجيه الرسائل القصيرة وتسليمها. عندما تعمل البوابة (العملاء) ESMEs الاتصالات من الخارجية. تقبل SMSC الاتصالات من ESMEs في **وضع الخادم**، تعمل كـ

SSL/TLS بروتوكولات التشفير للاتصالات الآمنة.

Submit_SM PDU SMPP لتقديم رسالة للتسليم.

Submit_SM_Resp استجابة SMPP لـ submit_sm، تشير إلى النجاح أو الفشل.

System ID اسم المستخدم المستخدم لمصادقة SMPP.

T

Telemetry جمع ونقل تلقائي لمقاييس النظام.

TON (نوع الرقم) حقل SMPP (على سبيل المثال، دولي، وطني) يحدد تنسيق الرقم.

TPS (المعاملات في الثانية) حد معدل أقصى للرسائل في الثانية عبر اتصال.

Transceiver نوع ارتباط SMPP (الأكثر شيوعًا) يمكنه إرسال واستقبال الرسائل.

Transmitter نوع ارتباط SMPP يرسل الرسائل فقط (submit_sm).

Throughput معدل معالجة الرسائل، يُقاس عادةً بالرسائل في الثانية.

U

UCS-2 ترميز Unicode 16 بت للرسائل القصيرة، يدعم حتى 70 حرفًا لكل رسالة.

Uptime مدة تشغيل الاتصال أو الخدمة بشكل مستمر.

V

Validity Period الحد الزمني لمحاولة تسليم الرسالة قبل انتهاء الصلاحية.

W

Web Dashboard واجهة مستخدم قائمة على المتصفح لمراقبة وإدارة البوابة.

Whitelist انظر قائمة IP البيضاء وقائمة عناوين المصدر البيضاء.

X

(لا توجد مصطلحات)

Y

(لا توجد مصطلحات)

Z

(لا توجد مصطلحات)

مرجع سريع للاختصارات

الاختصار	المصطلح الكامل
API	واجهة برمجة التطبيقات
CIDR	توجيه بين المجالات بدون فئات
DLR	إيصال التسليم
ESME	كيان الرسائل القصيرة الخارجي
GSM	النظام العالمي للاتصالات المحمولة
HTTP	بروتوكول نقل النصوص التشعبية
HTTPS	بروتوكول نقل النصوص التشعبية الآمن
IP	بروتوكول الإنترنت
ISDN	شبكة الخدمات الرقمية المتكاملة
KPI	مؤشر الأداء الرئيسي
MO	الرسائل المنشأة من الهاتف المحمول
MSISDN	رقم الدليل الدولي للمشترك في المحطة المحمولة
MT	الرسائل الموجهة إلى الهاتف المحمول
NPI	مؤشر خطة الترقيم
PDU	وحدة بيانات البروتوكول
SMPP	الرسائل القصيرة من نظير إلى نظير
SMSC	مركز خدمة الرسائل القصيرة

الاختصار	المصطلح الكامل
SMS	خدمة الرسائل القصيرة
SSL	طبقة المقابس الآمنة
TLS	أمان طبقة النقل
TON	نوع الرقم
TPS	المعاملات في الثانية
UCS	مجموعة الأحرف المشفرة العالمية
UI	واجهة المستخدم
URL	محدد موقع الموارد الموحد

الوثائق ذات الصلة

- **README.md** - نظرة عامة على النظام وكيفية البدء
- **CONFIGURATION.md** - شرح معلمات التكوين
- **USAGE.md** - العمليات اليومية
- **MONITORING.md** - المقاييس والمراقبة
- **TROUBLESHOOTING.md** - حل المشكلات

ذاكرة تخزين رسائل SMPP

نظرة عامة

ذاكرة تخزين رسائل SMPP هي طبقة تخزين محلية تسمح لبوابة SMPP بالاستمرار في قبول الرسائل الواردة حتى عندما تكون واجهة برمجة التطبيقات الخلفية غير متاحة. يتم تخزين الرسائل محليًا في Mnesia، ويتم تسليمها تلقائيًا إلى واجهة برمجة التطبيقات عند استعادة الاتصال، باستخدام منطق إعادة المحاولة الذكي مع زيادة زمن الانتظار بشكل أسي.

الميزات

- **قبول الرسائل المرن** - الاستمرار في قبول رسائل SMPP خلال انقطاع واجهة برمجة التطبيقات
- **تخزين دائم** - يستخدم Mnesia مع `:disc_copies` لضمان الاستمرارية عبر إعادة التشغيل
- **إعادة المحاولة التلقائية** - تحاول العمال في الخلفية التسليم تلقائيًا مع زيادة زمن الانتظار بشكل أسي
- **تكوين لكل ارتباط** - تمكين/تعطيل التخزين المؤقت بشكل مستقل لكل ارتباط SMPP
- **حماية من الفيضانات** - إخلاء FIFO عند وصول الذاكرة المؤقتة إلى حد الحجم المحدد
- **احتفاظ بالرسائل الفاشلة** - الاحتفاظ بالرسائل التي فشلت بشكل دائم للمراجعة اليدوية
- **المراقبة في الوقت الحقيقي** - لوحة معلومات LiveView مع إحصائيات الذاكرة المؤقتة والقياسات
- **قياسات Prometheus** - تصدير كامل للقياسات للمراقبة والتنبيه

الهيكلية

تدفق الرسائل

مع تمكين التخزين المؤقت

```
SMPP Client → submit_sm → Gateway Server
                                ↓
                        Cache in Mnesia (immediate response)
                                ↓
                        CacheFlushWorker (background)
                                ↓
                          Backend API
```

مع تعطيل التخزين المؤقت

```
SMPP Client → submit_sm → Gateway Server
                                ↓
                          Backend API (direct)
```

المكونات

1. **وحدة MessageCache**
    - منطق التخزين الأساسي
    - معالجة الفيضانات
    - وظائف الاستعلام لـ LiveView والعمال
2. **CacheFlushWorker**
    - GenServer لكل ارتباط مع تمكين التخزين المؤقت
    - يستعلم عن الرسائل الجاهزة لإعادة المحاولة
    - ينفذ زيادة زمن الانتظار بشكل أسي
    - يحدد الرسائل التي فشلت بشكل دائم
3. **جدول Mnesia** (`:smpp_message_cache`)
    - تخزين دائم مع `:disc_copies`
    - مفهرس بواسطة bind_name و next_retry_at و status
    - يبقى بعد إعادة تشغيل التطبيق

التكوين

الإعدادات العالمية

قم بتحرير `config/runtime.exs`:

```
config :omnimessage_smpp,
  # مدى تكرار استعلام العمال عن الرسائل (بالملي ثانية)
  cache_flush_interval: 10_000,

  # الحد الأقصى لعدد محاولات إعادة المحاولة قبل وضع علامة كفشل دائم
  cache_max_retry_attempts: 10,

  # مضاعف زيادة زمن الانتظار
  cache_backoff_multiplier: 2,

  # نوع تخزين Mnesia (:disc_copies أو :ram_copies)
  mnesia_storage_type: :disc_copies
```

تكوين لكل ارتباط

يمكن تكوين كل ارتباط SMPP (عميل أو خادم) بشكل مستقل:

```
config :omnimessage_smpp, :binds, [
  %{
    name: "my_smpp_bind",
    mode: :server,
    system_id: "username",
    password: "password",

    # تكوين التخزين المؤقت
    cache_enabled: true,         # تمكين التخزين المؤقت (افتراضي:
false)
    cache_max_size: 10_000,      # الحد الأقصى للرسائل للتخزين
(10,000 :افتراضي) المؤقت
    cache_retry_interval: 60     # فترة إعادة المحاولة الأساسية
(60 :افتراضي) بالثواني
  }
]
```

المتغيرات البيئية

```
# إعدادات التخزين المؤقت العالمية
CACHE_FLUSH_INTERVAL=10000              # فترة استعلام الإخلاء (مللي
ثانية)
CACHE_MAX_RETRY_ATTEMPTS=10             # الحد الأقصى لإعادة
المحاولات قبل الفشل الدائم
CACHE_BACKOFF_MULTIPLIER=2              # مضاعف زيادة زمن الانتظار
MNESIA_STORAGE_TYPE=disc_copies         # نوع تخزين Mnesia
```

سلوك إعادة المحاولة

زيادة زمن الانتظار بشكل أسي

عند فشل تسليم الرسالة، يتضاعف زمن إعادة المحاولة مع كل محاولة:

```
Base interval: 60 seconds
Backoff multiplier: 2

Retry 0: 60s   (1 minute)
Retry 1: 120s  (2 minutes)
Retry 2: 240s  (4 minutes)
Retry 3: 480s  (8 minutes)
Retry 4: 960s  (16 minutes)
Retry 5: 1920s (32 minutes)
...
Retry 9: 30,720s (8.5 hours)
```

الفشل الدائم

بعد 10 محاولات فاشلة (بشكل افتراضي)، يتم وضع علامة على الرسائل كـ `failed_permanent` و:

- تبقى في الذاكرة المؤقتة للمراجعة اليدوية
- تتوقف عن إعادة المحاولة تلقائيًا
- تظهر في قسم "الفشل الدائم" في لوحة معلومات الذاكرة المؤقتة
- يمكن إعادة المحاولة يدويًا أو مسحها

انتقالات الحالة

```
:pending → :delivering → SUCCESS (deleted from cache)
                       → FAILURE → :pending (retry with backoff)
                                 → :failed_permanent (after max
retries)
```

المراقبة

LiveView لوحة معلومات

يمكن الوصول إلى لوحة معلومات الذاكرة المؤقتة على `http://your-server:4000/smpp` ← علامة "ذاكرة الرسائل"

بطاقات الملخص:

- إجمالي التخزين المؤقت - جميع الرسائل الموجودة حاليًا في الذاكرة المؤقتة
- التسليم المعلق - الرسائل التي تنتظر إعادة المحاولة
- الفشل الدائم - الرسائل التي تجاوزت الحد الأقصى لإعادة المحاولات

جدول لكل ارتباط:

- اسم الارتباط
- عدد الرسائل المخزنة مؤقتًا
- تقسيم المعلق / الفاشل
- الحد الأقصى لحجم الذاكرة المؤقتة
- نسبة الاستخدام (مع شريط تقدم مرئي)

قياسات Prometheus

```
# حجم الذاكرة المؤقتة الحالية لكل ارتباط
smpp_cache_size{bind_name="my_bind",mode="server"} 42

# إجمالي الرسائل التي تم تسليمها بنجاح
smpp_cache_delivered_total{bind_name="my_bind"} 1234

# إجمالي محاولات إعادة المحاولة
smpp_cache_retry_total{bind_name="my_bind"} 56

# إجمالي الفشل الدائم
smpp_cache_permanent_failures_total{bind_name="my_bind"} 2

# (الرسائل المفقودة) إجمالي أحداث الفيضانات
smpp_cache_overflow_total{bind_name="my_bind"} 0
```

رسائل السجل

```
INFO  Message -123456789 cached for my_smpp_bind
INFO  Successfully delivered cached message -123456789, API ID:
99999
WARN  Failed to deliver message -123456789 (retry 3/10), next
retry at 2026-02-01 12:34:56Z
ERROR Message -123456789 exceeded max retries (10), marking as
failed_permanent
WARN  Cache overflow for my_smpp_bind: deleted oldest message
```

العمليات

تمكين/تعطيل التخزين المؤقت

عبر واجهة LiveView

1. انتقل إلى `http://your-server:4000/smpp`
2. انتقل إلى علامة "أقران العملاء" أو "أقران الخادم"
3. حرر الارتباط

4. قم بتبديل مربع "تمكين التخزين المؤقت"
5. احفظ التغييرات

عبر وحدة تحكم IEx

```
# تمكين التخزين المؤقت لارتباط
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: true,
  cache_max_size: 10_000,
  cache_retry_interval: 60
)

# تعطيل التخزين المؤقت
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: false
)
```

مراقبة حالة الذاكرة المؤقتة

```
# الحصول على ملخص عالمي
SmsC.SMPP.MessageCache.get_cache_summary()
# => %{total_cached: 42, pending: 40, failed: 2}

# الح��ول على تقسيم لكل ارتباط
SmsC.SMPP.MessageCache.get_cache_by_bind()
# => [
#   %{bind_name: "bind1", total: 30, pending: 28, failed: 2,
max_size: 10_000},
#   %{bind_name: "bind2", total: 12, pending: 12, failed: 0,
max_size: 10_000}
# ]

# عد الرسائل لارتباط محدد
SmsC.SMPP.MessageCache.count_cached_messages("my_bind")
# => 42
```

التدخلات اليدوية

مسح الرسائل الفاشلة

```
# الحصول على جميع الرسائل الفاشلة لارتباط
{:atomic, failed_messages} = :mnesia.transaction(fn ->
  :mnesia.match_object({:smpp_message_cache, :_, "my_bind", :_,
:_, :_, :_, :_, :_, :failed_permanent})
end)

# حذفها
Enum.each(failed_messages, fn {_, {bind_name, msg_id}, _, _, _, _,
_, _, _, _} ->
  SmsC.SMPP.MessageCache.delete_cache_record(bind_name, msg_id)
end)
```

فرض إعادة المحاولة لرسالة فاشلة

```
# إعادة تعيين رسالة failed_permanent إلى pending
SmsC.SMPP.MessageCache.update_cache_record("my_bind", -123456, %{
  status: :pending,
  retry_count: 0,
  next_retry_at: DateTime.utc_now(),
  last_error: nil
})
```

استكشاف الأخطاء وإصلاحها

الذاكرة المؤقتة ممتلئة / أحداث الفيضانات

الأعراض: زيادة قياس `cache_overflow_total`، حذف الرسائل الأقدم

السبب: تم الوصول إلى حد حجم الذاكرة المؤقتة

الحلول:

1. زيادة `cache_max_size` للارتباط

2. التحقيق في سبب فشل تسليم واجهة برمجة التطبيقات (تحقق من سجلات واجهة برمجة التطبيقات، الشبكة)
3. مسح الرسائل القديمة الفاشلة يدويًا
4. تحقق مما إذا كانت فترة الإخلاء بطيئة جدًا

الرسائل لا يتم تسليمها

الأعراض: الرسائل عالقة في حالة `:pending`

الأسباب المحتملة:

1. **واجهة برمجة التطبيقات معطلة**
   - تحقق من توفر واجهة برمجة التطبيقات
   - تحقق من سجلات واجهة برمجة التطبيقات الخلفية
   - تحقق من اتصال الشبكة
2. **next_retry_at في المستقبل**
   - سيتم إعادة محاولة الرسائل عند الوصول إلى `next_retry_at`
   - تحقق من جدول زيادة زمن الانتظار بشكل أسي
3. **عامل الإخلاء لا يعمل**

   ```
   # تحقق مما إذا كانت العمال تعمل
   Supervisor.which_children(SmsC.SMPP.Supervisor)
   ```

4. **التخزين المؤقت معطل**
   - تحقق من `cache_enabled: true` في تكوين الارتباط

عدد المحاولات العالية

الأعراض: العديد من الرسائل مع قيم `retry_count` عالية

التحقيق:

```
# العثور على الرسائل ذات عدد المحاولات العالية
{:atomic, messages} = :mnesia.transaction(fn ->
  :mnesia.match_object({:smpp_message_cache, :_, "my_bind", :_,
:_, :_, :_, :_, :_, :_})
end)

high_retry = Enum.filter(messages, fn {_, _, _, _, _, _,
retry_count, _, _, _} ->
  retry_count >= 5
end)

# تحقق من last_error لكل منها
Enum.each(high_retry, fn {_, _, _, msg_id, _, _, retry_count, _,
last_error, _} ->
  IO.puts("Message #{msg_id}: #{retry_count} retries, error: #
{inspect(last_error)}")
end)
```

مساحة قرص Mnesia

الأعراض: امتلاء مساحة القرص

تحقق من دليل Mnesia:

```
ls -lh Mnesia.*
du -sh Mnesia.*
```

تنظيف:

1. مسح الرسائل القديمة الفاشلة (انظر التدخلات اليدوية أعلاه)
2. تقليل `cache_max_size` لكل ارتباط
3. تمكين الفيضانات (المناسب FIFO ضمان الإخلاء)

اعتبارات الأداء

استخدام الذاكرة

- تستخدم كل رسالة مخزنة مؤقتًا حوالي 500-1000 بايت (اعتمادًا على حجم الرسالة)
- 10,000 رسالة ≈ 5-10 ميغابايت من الذاكرة
- يتم أيضًا كتابة البيانات إلى القرص، مع `:disc_copies`

استخدام وحدة المعالجة المركزية

- يستعلم عمال الإخلاء كل 10 ثوانٍ بشكل افتراضي (يمكن تكوينه)
- المعالجة الدفعة (100 رسالة لكل دورة) تقلل من الحمل الزائد
- التسليم المتزامن (حد أقصى 10 مكالمات واجهة برمجة التطبيقات المتزامنة لكل عامل)

إدخال/إخراج القرص

- `:disc_copies` تكتب إلى القرص في كل معاملة
- من أجل معدل نقل بيانات مرتفع جدًا (>1000 msg/sec)، ضع في اعتبارك:
  - استخدام `:ram_copies` (يفقد الاستمرارية)
  - زيادة فترات الإخلاء
  - التوسع أفقيًا

الحدود الموصى بها

السيناريو	cache_max_size	cache_flush_interval
حجم منخفض (<100 msg/sec)	10,000	10,000ms
حجم متوسط (100-500 msg/sec)	50,000	5,000ms
حجم مرتفع (>500 msg/sec)	100,000	3,000ms

سيناريوهات الاسترداد

إعادة تشغيل التطبيق

1. يقوم Mnesia بتحميل جداول `:disc_copies` من القرص تلقائيًا
2. تبقى الرسائل المخزنة مؤقتًا سليمة
3. يتم إعادة تشغيل عمال الإخلاء ويستمرون في المعالجة

ترحيل قاعدة البيانات

عند الترقية من إصدار بدون دعم التخزين المؤقت:

1. يضيف الترحيل تلقائيًا حقول التخزين المؤقت إلى الروابط الموجودة
2. القيم الافتراضية: `cache_enabled: false`، `cache_max_size: 10_000`، `cache_retry_interval: 60`
3. لا فقدان للبيانات
4. يتم إنشاء جدول التخزين المؤقت عند التشغيل الأول

استرداد من انقطاع واجهة برمجة التطبيقات

1. تتجمع الرسائل في الذاكرة المؤقتة أثناء الانقطاع
2. عند استعادة واجهة برمجة التطبيقات، يقوم عمال الإخلاء بالتسليم تلقائيًا
3. يتم تسليم الرسائل الأقدم أولاً (FIFO)
4. تمنع زيادة زمن الانتظار ب��كل أسي من تحميل واجهة برمجة التطبيقات أثناء الاسترداد

أفضل الممارسات

1. **تمكين التخزين المؤقت بشكل افتراضي** - يمنع فقدان الرسائل أثناء الانقطاعات
2. **مراقبة القياسات** - إعداد تنبيهات على `cache_permanent_failures_total` و `cache_overflow_total`
3. **تحديد الحجم بشكل مناسب** - تعيين `cache_max_size` بناءً على مدة الانقطاع المتوقعة
4. **مراجعة الرسائل الفاشلة** - التحقق بانتظام من الرسائل الفاشلة الدائمة للأنماط

5. **اختبار الفشل** - محاكاة انقطاعات واجهة برمجة التطبيقات للتحقق من سلوك التخزين المؤقت
6. **تعديل فترات إعادة المحاولة** - ضبطها بناءً على أنماط وقت استعادة واجهة برمجة التطبيقات
7. **استخدام التخزين الدائم** `mnesia_storage_type: disc_copies` - الاحتفاظ بـ في الإنتاج

انظر أيضًا

- مرجع التكوين
- المراقبة والقياسات
- استكشاف الأخطاء وإصلاحها

دليل المراقبة والقياسات

مرجع كامل لمراقبة بوابة SMPP

نظرة عامة

تقوم بوابة SMPP بعرض القياسات بتنسيق Prometheus لمراقبة صحة الاتصال، وسرعة الرسائل، وأداء النظام.

حرج: نظرًا لأن البوابة بدون حالة وتعتمد على OmniMessage Core، **فإن الاتصال بـ OmniMessage هو أهم مقياس يجب مراقبته**. راقب كلا من:

1. **قياسات بوابة SMPP** - صحة مستوى البروتوكول
2. **قياسات OmniMessage API** - الاتصال والصحة في الخلفية

نقطة نهاية القياسات

URL: `http://your-server:4000/metrics`

التنسيق: تنسيق نص Prometheus

الوصول: مفتوح لـ localhost (قم بتكوين جدار الحماية للوصول عن بُعد) بشكل افتراضي

اختبار سريع

```
curl http://localhost:4000/metrics
```

القياسات الم��احة

تبدأ جميع القياسات بـ `smpp_` وتتضمن تسميات للتعريف.

قياسات الترخيص

omnimessage_smpp_license_status

النوع: Gauge
الوصف: حالة الترخيص الحالية
القيم:

- `1` = ترخيص صالح
- `0` = ترخيص غير صالح/منتهي

التسميات: لا شيء

مثال:

```
omnimessage_smpp_license_status 1
```

الاستخدام:

- تنبيه عندما تكون القيمة 0 (ترخيص غير صالح)
- عندما يكون الترخيص غير صالح، تتوقف معالجة قائمة الانتظار الصادرة ولكن تبقى اتصالات SMPP متصلة
- تظل واجهة الويب متاحة لاستكشاف الأخطاء وإصلاحها

اسم المنتج: `omnimessage_smpp`

ملاحظات:

- تتوقف البوابة عن معالجة قوائم الانتظار الصادرة، عندما يكون الترخيص غير صالح (`license_status == 0`)
- تبقى اتصالات SMPP متصلة وتقبل طلبات الربط (كل من العميل والخادم)
- لا تزال الرسائل الواردة تُستقبل ولكن لا تتم معالجتها
- تظل واجهة المستخدم والمراقبة متاحة بغض النظر عن ح��لة الترخيص

مثال على التنبيه:

```
- alert: SMPP_License_Invalid
  expr: omnimessage_smpp_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "ترخيص بوابة SMPP غير صالح أو منتهي"
    description: "حالة الترخيص غير صالحة - معالجة الرسائل الصادرة محظورة"
```

قياسات حالة الاتصال

smpp_connection_status

النوع: Gauge
الوصف: حالة الاتصال الحالية لربط SMPP
القيم:

- `1` = متصل. بالنسبة لربط الخادم، يجب أن تكون هناك جلسة ESME واحدة على الأقل متصلة.
- `0` = غير متصل. بالنسبة لربط الخادم، لا توجد جلسة ESME متصلة.

التسميات:

- `bind_name` - اسم الاتصال (مثل "vodafone_uk")
- `mode` - نوع الاتصال ("server" أو "client")
- `host` - المضيف البعيد (وضع العميل فقط)
- `port` - المنفذ البعيد (وضع العميل فقط)
- `bind_type` - نوع ربط SMPP (وضع العميل فقط)
- `system_id` - معرف النظام المستخدم

مثال:

```
smpp_connection_status{bind_name="vodafone_uk",mode="client",host="sm
1
```

الاستخدام:

- تنبيه عندما تكون القيمة 0 (غير متصل)
- تتبع نسبة وقت الاتصال
- مراقبة تكرار إعادة الاتصال

عدادات الرسائل

smpp_messages_sent_total

النوع: Counter
الوصف: إجمالي عدد الرسائل المرسلة عبر ربط SMPP: `deliver_sm` إلى ESME على ربط الخادم، `submit_sm` إلى الناقل على ربط العميل
الوحدة: PDUs SMPP. كل جزء من رسالة متعددة الأجزاء يُحتسب مرة واحدة.

التسميات: نفس تسميات حالة الاتصال

مثال:

```
smpp_messages_sent_total{bind_name="vodafone_uk",mode="client",...}
150234
```

الاستخدام:

- حساب معدل الرسائل (رسائل/ثانية)
- تتبع الحجم اليومي/الشهري
- مقارنة الإنتاج الفعلي مقابل المتوقع

smpp_messages_received_total

النوع: Counter
الوصف: إجمالي عدد الرسائل المستلمة عبر ربط SMPP: `submit_sm` من ESME على ربط الخادم، `deliver_sm` من الناقل على ربط العميل. لا تُحتسب إيصا��ات التسليم.
الوحدة: PDUs SMPP

التسميات: نفس تسميات حالة الاتصال

مثال:

```
smpp_messages_received_total{bind_name="partner_acme",mode="server",..
45123
```

الاستخدام:

- مراقبة حجم الرسائل الواردة
- تتبع حركة المرور المرسلة من الهواتف المحمولة (MO)
- تنبيه على تغييرات غير متوقعة في الحجم

قياسات التسليم

smpp_delivery_failures_total

النوع: Counter
الوصف: إجمالي عدد حالات فشل تسليم الرسائل `deliver_sm` أو `submit_sm` التي لم يمكن إرسالها، أو التي تم الرد عليها بحالة غير صفرية
الوحدة: PDUs SMPP

التسميات: نفس تسميات حالة الاتصال

مثال:

```
smpp_delivery_failures_total{bind_name="vodafone_uk",mode="client",..
234
```

الاستخدام:

- حساب معدل نجاح التسليم
- تنبيه على معدلات الفشل العالية
- تحديد الاتصالات المشكلة

حساب معدل النجاح:

```
success_rate = (messages_sent - delivery_failures) / messages_sent
* 100
```

قياسات عمليات الربط

smpp_bind_success_total

النوع: Counter
الوصف: إجمالي عدد عمليات الربط الناجحة
الوحدة: محاولات الربط

مثال:

```
smpp_bind_success_total{bind_name="vodafone_uk",...} 45
```

الاستخدام:

- تتبع استقرار الربط
- مراقبة نجاح المصادقة

smpp_bind_failures_total

النوع: Counter
الوصف: إجمالي عدد عمليات الربط الفاشلة
الوحدة: محاولات الربط

مثال:

```
smpp_bind_failures_total{bind_name="vodafone_uk",...} 3
```

الاستخدام:

- تنبيه على فشل المصادقة
- تحديد مشاكل الاعتماد
- تتبع مشاكل الاتصال بالناقل

قياسات أحداث الاتصال

smpp_connection_attempts_total

النوع: Counter
الوصف: إجمالي عدد محاولات الاتصال. على ربط الخادم، عدد جلسات ESME التي تم ربطها.
الوحدة: محاولات

مثال:

```
smpp_connection_attempts_total{bind_name="vodafone_uk",...} 48
```

الاستخدام:

- تتبع دوران الاتصال
- مراقبة تكرار إعادة الاتصال

smpp_disconnection_total

النوع: Counter
الوصف: إجمالي عدد حالات الانفصال. على ربط الخادم، عدد جلسات ESME المرتبطة التي انتهت.
الوحدة: حالات الانفصال

مثال:

```
smpp_disconnection_total{bind_name="vodafone_uk",...} 3
```

الاستخدام:

- تنبيه على حالات الانفصال المتكررة
- تحديد مشاكل الشبكة
- تتبع استقرار الاتصال

smpp_reconnect_attempts_total

النوع: Counter
الوصف: إجمالي عدد محاولات إعادة الاتصال المجدولة لربط العميل بعد فشل الاتصال، أو رفض الربط، أو جلسة مفقودة
الوحدة: محاولات إعادة الاتصال

مثال:

```
smpp_reconnect_attempts_total{bind_name="vodafone_uk",...} 7
```

الاستخدام:

- اكتشاف ناقل يقبل الاتصال ثم يقطعه على الفور (عداد يرتفع بسرعة بينما تبقى `smpp_connection_status` على `0`)
- قياس المدة التي قضتها عملية الربط في محاولة العودة بعد انقطاع

قياسات استفسار الرابط

smpp_enquire_link_sent_total

النوع: Counter
الوصف: إجمالي عدد PDUs استفسار الرابط المرسلة للتحقق من حيوية الاتصال
الوحدة: PDUs

التسميات: نفس تسميات حالة الاتصال

مثال:

```
smpp_enquire_link_sent_total{bind_name="vodafone_uk",mode="client",..
1440
```

الاستخدام:

- تتبع نشاط إبقاء الاتصال حياً
- المقارنة مع المستلم لاكتشاف الفشل من جانب واحد

smpp_enquire_link_received_total

النوع: Counter
الوصف: إجمالي عدد PDUs استجابة استفسار الرابط المستلمة من النظير البعيد
الوحدة: PDUs

التسميات: نفس تسميات حالة الاتصال

مثال:

```
smpp_enquire_link_received_total{bind_name="vodafone_uk",mode="client
1438
```

الاستخدام:

- اكتشاف النظائر غير المستجيبة (المُرسل >> المستلم)
- مراقبة صحة الاتصال بما يتجاوز الحالة البسيطة

قياسات وقت التشغيل

smpp_uptime_seconds

النوع: Gauge
الوصف: وقت التشغيل الحالي لربط SMPP بالثواني
الوحدة: ثواني

مثال:

```
smpp_uptime_seconds{bind_name="vodafone_uk",...} 86400
```

الاستخدام:

- تتبع استقرار الاتصال
- حساب نسبة وقت التشغيل
- تنبيه على إعادة التشغيل الأخيرة

قياسات ذاكرة التخزين المؤقت للرسائل

تكون هذه القياسات متاحة عندما تكون ذاكرة التخزين المؤقت للرسائل مفعلة على واحد أو أكثر من الروابط. انظر MESSAGE_CACHE.md لتفاصيل تكوين الذاكرة المؤقتة.

smpp_cache_size

النوع: Gauge
الوصف: العدد الحالي للرسائل في الذاكرة المؤقتة المحلية لكل ربط
الوحدة: رسائل

التسميات:

- `bind_name` - اسم الاتصال
- `mode` - نوع الاتصال ("server" أو "client")

مثال:

```
smpp_cache_size{bind_name="partner_acme",mode="server"} 42
```

الاستخدام:

- مراقبة استخدام الذاكرة المؤقتة
- تنبيه عند الاقتراب من `cache_max_size`

smpp_cache_delivered_total

النوع: Counter
الوصف: إجمالي عدد الرسائل المخزنة مؤقتًا التي تم تسليمها بنجاح إلى واجهة برمجة التطبيقات الخلفية
الوحدة: رسائل

مثال:

```
smpp_cache_delivered_total{bind_name="partner_acme"} 1234
```

smpp_cache_retry_total

النوع: Counter
الوصف: إجمالي عدد محاولات إعادة الإرسال للرسائل المخزنة مؤقتًا
الوحدة: محاولات

مثال:

```
smpp_cache_retry_total{bind_name="partner_acme"} 56
```

smpp_cache_permanent_failures_total

النوع: Counter
الوصف: إجمالي عدد الرسائل التي تجاوزت الحد الأقصى لمحاولات إعادة الإرسال وتم وضع علامة عليها على أنها فشلت بشكل دائم
الوحدة: رسائل

مثال:

```
smpp_cache_permanent_failures_total{bind_name="partner_acme"} 2
```

الاستخدام:

- تنبيه عند > 0 (يتطلب مراجعة يدوية)

smpp_cache_overflow_total

النوع: Counter
الوصف: إجمالي عدد أحداث تجاوز الذاكرة المؤقتة حيث تم طرد أقدم رسالة لإفساح المجال
الوحدة: أحداث

مثال:

```
smpp_cache_overflow_total{bind_name="partner_acme"} 0
```

الاستخدام:

- تنبيه عند الزيادة (الذاكرة المؤقتة صغيرة جدًا أو انقطاع واجهة برمجة التطبيقات طويل جدًا)

قياسات إيصال التسليم (DLR)

تكون هذه القياسات متاحة لعمليات الربط الخادمة التي تحدد `deliver_receipts: true`.
انظر CONFIGURATION.md للإعداد و USAGE.md لتدفق الرسائل.

smpp_dlr_tracked_total

النوع: Counter
الوصف: إجمالي عدد الرسائل المرسلة المسجلة لتتبع إيصال التسليم
الوحدة: رسائل

التسميات:

- `bind_name` - اسم الاتصال
- `mode` - نوع الاتصال ("server" أو "client")

مثال:

```
smpp_dlr_tracked_total{bind_name="masorange",mode="server"} 10234
```

smpp_dlr_sent_total

النوع: Counter
الوصف: إجمالي عدد إيصالات التسليم المرسلة مرة أخرى إلى الربط المرسل
الوحدة: إيصالات

مثال:

```
smpp_dlr_sent_total{bind_name="masorange",mode="server"} 10180
```

الاستخدام:

- لرؤية نسبة إكمال الإيصال `smpp_dlr_tracked_total` المقارنة مع

smpp_dlr_dropped_total

النوع: Counter
الوصف: إجمالي عدد الرسائل المتعقبة التي تم إسقاطها بدون إيصال. تقوم البوابة بإسقاط رسالة عندما لم يعد النواة لديها، أو عندما لا تصل الرسالة إلى حالة نهائية خلال 24 ساعة.
الوحدة: رسائل

مثال:

```
smpp_dlr_dropped_total{bind_name="masorange",mode="server"} 12
```

الاستخدام:

- تنبيه عند الزيادة (الرسائل لا تصل أبدًا إلى حالة نهائية، أو تفقدها النواة)

smpp_dlr_pending

النوع: Gauge
الوصف: العدد الحالي للرسائل المعلقة التي تنتظر حالة نهائية وإيصال
الوحدة: رسائل

مثال:

```
smpp_dlr_pending{bind_name="masorange",mode="server"} 37
```

الاستخدام:

- مراقبة تراكم المتعقب
- تنبيه عندما تبقى القيمة مرتفعة (النواة بطيئة في الوصول إلى الحالات النهائية، أو جلسة النظير متوقفة)

استعلامات المثال:

```
# نسبة إكمال الإيصال خلال 5 دقائق
sum(rate(smpp_dlr_sent_total[5m]))
  / sum(rate(smpp_dlr_tracked_total[5m]))

# إيصالات مسق��ة في الثانية
rate(smpp_dlr_dropped_total[5m])

# تراكم المتعقب الحالي لكل ربط
smpp_dlr_pending
```

قياسات صحة OmniMessage API

فإن صحة OmniMessage API، بينما تقوم البوابة نفسها بعرض قياسات متعلقة بـ SMPP **حرجة**. يجب عليك أيضًا مراقبة:

من قياسات OmniMessage (إذا كانت متاحة)

- `omnimessage_api_requests_total` - إجمالي طلبات API من البوابة
- `omnimessage_api_request_duration_seconds` - أوقات استجابة API
- `omnimessage_queue_depth` - الرسائل المعلقة في قائمة انتظار OmniMessage

من سجلات البوابة (إذا لم يتم عرض القياسات)

ابحث عن هذه الأنماط لاكتشاف مشاكل API:

- "api.*connection refused" - لا يمكن الوصول إلى OmniMessage
- "api.*timeout" - لا تستجيب OmniMessage
- "api.*http 503" - متوقفة مؤقتًا OmniMessage
- "api.*parse error" - مشكلة في تنسيق الاستجابة

تكوين Prometheus

تكوين سحب أساسي

أضف إلى `/etc/prometheus/prometheus.yml`:

```yaml
scrape_configs:
  - job_name: 'omnimessage-smpp'
    scrape_interval: 15s
    static_configs:
      - targets: ['your-server:4000']
        labels:
          environment: 'production'
          service: 'omnimessage-smpp'
```

بوابات متعددة

```yaml
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    scrape_interval: 15s
    static_configs:
      - targets:
        - 'smpp-gw-1:4000'
        - 'smpp-gw-2:4000'
        - 'smpp-gw-3:4000'
        labels:
          environment: 'production'
```

اكتشاف الخدمة

باستخدام اكتشاف قائم على الملفات:

```yaml
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    file_sd_configs:
      - files:
        - '/etc/prometheus/targets/smpp-*.json'
```

ملف `/etc/prometheus/targets/smpp-production.json`:

```
[
  {
    "targets": ["smpp-gw-1:4000", "smpp-gw-2:4000"],
    "labels": {
      "environment": "production",
      "datacenter": "us-east"
    }
  }
]
```

لوحات معلومات Grafana

لوحات معلومات نموذجية

لوحة حالة الاتصال

الاستعلام:

```
smpp_connection_status{job="omnimessage-smpp"}
```

التصور: Stat
الحدود:

- أحمر: القيمة < 1 (غير متصل)
- أخضر: القيمة == 1 (متصل)

لوحة معدل الرسائل

الاس��علام:

```
rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
```

التصور: Graph
الوحدة: رسائل/ثانية
الأسطورة: `{{bind_name}}`

لوحة معدل نجاح التسليم

الاستعلام:

```
100 * (1 - (
  rate(smpp_delivery_failures_total{job="omnimessage-smpp"}[5m])
  /
  rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
))
```

التصور: Gauge
الوحدة: النسبة المئوية (0-100)
الحدود:

- أحمر: < 95%
- أصفر: 95-98%
- أخضر: > 98%

لوحة وقت تشغيل الاتصال

الاستعلام:

```
smpp_uptime_seconds{job="omnimessage-smpp"} / 3600
```

التصور: Stat
الوحدة: ساعات

قواعد التنبيه

قواعد تنبيه Prometheus

احفظ إلى `/etc/prometheus/rules/smpp-alerts.yml`:

```yaml
groups:
  - name: smpp_gateway
    interval: 30s
    rules:
      # الاتصال متوقف
      - alert: SMPPConnectionDown
        expr: smpp_connection_status == 0
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: "الاتصال SMPP {{ $labels.bind_name }} متوقف"
          description: "الاتصال {{ $labels.bind_name }} متوقف لأكثر من دقيقتين."

      # معدل فشل مرتفع
      - alert: SMPPHighFailureRate
        expr: |
          (
            rate(smpp_delivery_failures_total[5m])
            /
            rate(smpp_messages_sent_total[5m])
          ) > 0.05
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "معدل فشل تسليم مرتفع على {{ $labels.bind_name }}"
          description: "معدل فشل التسليم هو {{ $value | humanizePercentage }} على {{ $labels.bind_name }}."

      # فشل الربط
      - alert: SMPPBindFailures
        expr: increase(smpp_bind_failures_total[10m]) > 3
        labels:
          severity: warning
        annotations:
          summary: "فشل ربط متعدد على {{ $labels.bind_name }}"
          description: "{{ $labels.bind_name }} فشلت في الربط {{ $value }} مرات في آخر 10 دقائق."

      # لا توجد رسائل مرسلة (عند المتوقع)
```

```
      - alert: SMPPNoTraffic
        expr: rate(smpp_messages_sent_total[10m]) == 0
        for: 30m
        labels:
          severity: warning
        annotations:
          summary: "لا توجد رسائل مرسلة على {{ $labels.bind_name
}}"
          description: "لم ترسل أي رسائل {{ $labels.bind_name }}
لمدة 30 دقيقة."

      # انقطاعات متكررة
      - alert: SMPPFrequentDisconnections
        expr: increase(smpp_disconnection_total[1h]) > 5
        labels:
          severity: warning
        annotations:
          summary: "انقطاعات متكررة على {{ $labels.bind_name }}"
          description: "{{ $labels.bind_name }} انقطعت {{ $value
}} مرات في الساعة الماضية."

      # OmniMessage API غير متاحة
      - alert: OmniMessageAPIUnreachable
        expr: |
          count(count_over_time({job="omnimessage-smpp"} |=
"api.*connection refused"[5m])) > 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "OmniMessage API غير متاحة"
          description: "لا يمكن لـبوابة SMPP الوصول إلى OmniMessage
API. تحقق من تكوين API_BASE_URL والاتصال بالشبكة."

      # مهلات OmniMessage API
      - alert: OmniMessageAPITimeout
        expr: |
          count(count_over_time({job="omnimessage-smpp"} |=
"api.*timeout"[5m])) > 5
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: "OmniMessage API تتعرض لمهلات"
```

```
          description: "قد تكون API. تم اكت��اف مهلات متعددة في
OmniMessage بطيئة أو محملة بشكل زائد."

      # لا توجد تدفقات رسائل (مشكلة API)
      - alert: NoMessageFlow
        expr: rate(smpp_messages_sent_total[10m]) == 0 and
rate(smpp_messages_received_total[10m]) == 0
        for: 30m
        labels:
          severity: warning
        annotations:
          summary: "لا توجد تدفقات رسائل مكتشفة - تحقق من اتصال
OmniMessage"
          description: "لا توجد رسائل مرسلة أو مستلمة لمدة 30
دقيقة. تحقق من اتصال OmniMessage API وحالة قائمة الانتظار."
```

قم بتحميل القواعد في `prometheus.yml`:

```
rule_files:
  - '/etc/prometheus/rules/smpp-alerts.yml'
```

مراقبة لوحة المعلومات على الويب

تقدم واجهة الويب المدمجة مراقبة في الوقت الحقيقي بدون Prometheus.

الوصول

URL: `https://your-server:8087`

صفحة الحالة الحية

التنقل: SMPP → الحالة الحية

الميزات:

- حالة الاتصال في الوقت الحقيقي
- عدادات الرسائل

- وقت تشغيل الاتصال
- ناصر التحكم في إعادة الاتصال/الفصل اليدوي��
- تحديث تلقائي كل 5 ثوانٍ

الاستخدام:

- فحص الحالة السريعة
- التدخل اليدوي
- استكشاف الأخطاء وإصلاحها في الوقت الحقيقي

تظهر لوحة المعلومات:

- **إجمالي الربط**: العدد المجمع لجميع اتصالات العميل والخادم
- **ربط العملاء**: اتصالات صادرة إلى الناقلين (تظهر عدد المتصلين/غير المتصلين)
- **ربط الخوادم**: اتصالات واردة من الشركاء (تظهر العدد النشط/الانتظار)
- **استماع الخادم**: تكوين مقبس الخادم الوارد (المضيف، المنفذ، الحد الأقصى من الاتصالات)

مراقبة السجلات

سجلات النظام

عرض السجلات:

```
# متابعة السجلات في الوقت الحقيقي
sudo journalctl -u omnimessage-smpp -f

# آخر 100 سطر
sudo journalctl -u omnimessage-smpp -n 100

# منذ وقت محدد
sudo journalctl -u omnimessage-smpp --since "1 hour ago"

# تصفية حسب المستوى
sudo journalctl -u omnimessage-smpp -p err
```

سجلات واجهة الويب

التنقل: علامة التبويب السجلات في واجهة الويب

الميزات:

- تدفق السجلات في الوقت الحقيقي
- تصفية حسب المستوى (تصحيح، معلومات، تحذير، خطأ)
- البحث في السجلات
- إيقاف مؤقت/استئناف
- مسح السجلات

تتيح لك عرض السجلات:

- **تصفية المستوى**: اختر مستوى السجل (الكل، تصحيح، معلومات، تحذير، خطأ)
- **البحث**: العثور على إدخالات سجل محددة حسب محتوى النص
- **التمرير التلقائي**: تمكين/تعطيل التمرير التلقائي عند وصول سجلات جديدة
- **إيقاف مؤقت/استئناف**: إيقاف تحديثات السجل لمراجعة إدخالات معينة
- **مسح**: مسح جميع السجلات المعروضة

مؤشرات الأداء الرئيسية (KPIs)

صحة الاتصال

المقياس: نسبة وقت تشغيل الاتصال

```
avg_over_time(smpp_connection_status[24h]) * 100
```

الهدف: > 99.9%

معدل تسليم الرسائل

المقياس: الرسائل المرسلة في الثانية

```
rate(smpp_messages_sent_total[5m])
```

الهدف: يتطابق مع الحجم المتوقع

معدل نجاح التسليم

المقياس: نسبة التسليم الناجح

```
100 * (1 - rate(smpp_delivery_failures_total[5m]) /
rate(smpp_messages_sent_total[5m]))
```

الهدف: > 98%

استقرار الربط

المقياس: محاولات الربط في الساعة

```
rate(smpp_bind_success_total[1h]) * 3600
```

الهدف: < 10 في الساعة (يشير إلى اتصال مستقر)

أفضل ممارسات المراقبة

1. إعداد التنبيهات

- قم بتكوين تنبيهات Prometheus للمقاييس الحرجة
- استخدم PagerDuty/OpsGenie للتنبيه على مدار الساعة
- اختبر التنبيهات بانتظام

2. إنشاء لوحات المعلومات

- قم ببناء لوحات معلومات Grafana لكل بوابة

- تضمين جميع الاتصالات في لوحة معلومات واحدة
- إضافة لوحات تخطيط السعة

3. المراجعات المنتظمة

- مراجعة المقاييس أسبوعياً
- تحديد الاتجاهات والأنماط
- تخطيط تعديلات السعة

4. توثيق الأسس

- سجل أحجام الرسائل العادية
- وثق معدلات TPS المتوقعة
- لاحظ أوقات/أيام الذروة

5. الربط مع الخلفية

- راقب مقاييس API الخلفية
- تتبع تدفق الرسائل من البداية إلى النهاية
- تحديد الاختناقات

استكشاف الأخطاء باستخدام المقاييس

مشاكل الاتصال

تحقق: `smpp_connection_status`

- القيمة 0 = مراجعة السجلات، التحقق من الشبكة، التحقق من بيانات الاعتماد
- تغييرات متكررة = عدم استقرار الشبكة

معدلات التسليم الضعيفة

تحقق: `smpp_delivery_failures_total`

- معدل مرتفع = تحقق من حالة الناقل، مراجعة تنسيق الرسالة
- قارن عبر الاتصالات = تحديد الناقل المسبب للمشكلة

الإنتاجية المنخفضة

تحقق: معدل `smpp_messages_sent_total`

- توفر الطابور، TPS أقل من المتوقع = تحقق من حدود
- تحقق من مقاييس API الخلفية

مشاكل الربط

تحقق: `smpp_bind_failures_total`

- في تزايد = مشاكل في المصادقة، مشاكل في بيانات الاعتماد
- تحقق من system_id وكلمة المرور في التكوين

الوثائق ذات الصلة

- **CONFIGURATION.md** - تكوين إعدادات المراقبة
- **USAGE.md** - الإجراءات التشغيلية
- **TROUBLESHOOTING.md** - حل المشكلات
- **README.md** - نظرة عامة وبدء سريع

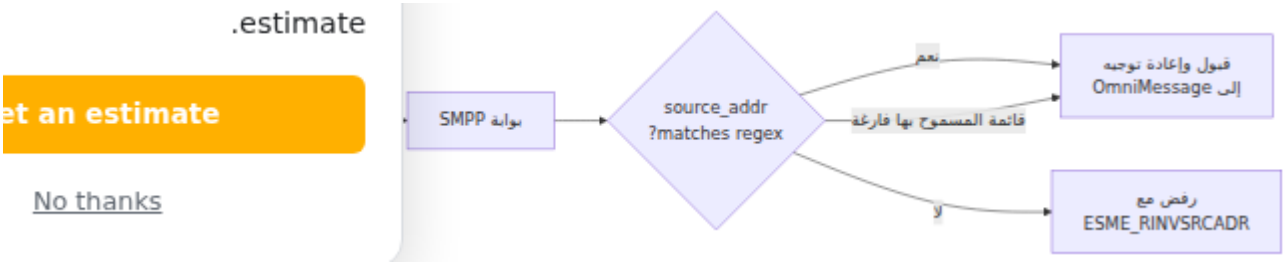
قائمة العناوين المسموح بها

التي يمكن أن يستخدمها (source_addr) تحكم لكل نظير في العناوين الأصلية عميل SMPP عند إرسال الرسائل.

نظرة عامة

عندما يرسل ESME (عميل) خارجي PDU `submit_sm` عبر بوابة SMPP، يتضمن PDU حقل `source_addr` الذي يمثل العنوان الأصلي (معرف المرسل / CLI). يمكن، بشكل افتراضي، للعملاء المعتمدين استخدام أي عنوان مصدر. تتيح ميزة قائمة العناوين المسموح بها للمشغلين تقييد العناوين المصدرية التي يُسمح لكل نظير خادم باستخدامها.

المسموح بها الحالية: عندما تكون القائمة فارغة، يُسمح IP يتبع هذا نفس النمط كما في قائمة بجميع القيم. عند تعبئتها، يتم قبول العناوين المصدرية المطابقة فقط.



قواعد المطابقة

كل إدخال في قائمة المسموح بها هو نمط **تعبير منتظم**. يتم اختبار العنوان المصدر ضد كل نمط. إذا تطابق أي نمط، يُسمح بالعنوان.

المطابقة الدقيقة

قم بتثبيت النمط باستخدام `^` و `$` لطلب مطابقة دقيقة. المطابقة هي **حساسة لحالة الأحرف** بشكل افتراضي.

إدخال قائمة المسموح بها	عنوان المصدر	النتيجة
`^MyBrand$`	`MyBrand`	مسموح
`^MyBrand$`	`mybrand`	مرفوض
`^MyBrand$`	`MyBrands`	مرفوض
`^\+61400000001$`	`+61400000001`	مسموح

مطابقة البادئة

استخدم `^` بدون `$` في النهاية لمطابقة العناوين التي تبدأ ببادئة.

إدخال قائمة المسموح بها	عنوان المصدر	النتيجة
`^614`	`61400000001`	مسموح
`^614`	`61412345678`	مسموح
`^614`	`61500000001`	مرفوض
`^\+614`	`+61400000001`	مسموح

الأنماط المتقدمة

يدعم بناء جملة التعبير المنتظم الكامل، مما يمكّن من قواعد مطابقة قوية.

إدخال قائمة المسموح بها	الوصف	الأمثلة
`^61[45]\d{8}$`	أرقام الهواتف المحمولة الأسترالية 614/615 (11 رقمًا)	`61400000001`, `61512345678`
`^(MyBrand|AlertSvc)$`	مطابقة دقيقة على أي من الاسمين	`MyBrand`, `AlertSvc`
`(?i)^mybrand$`	مطابقة دقيقة غير حساسة لحالة الأحرف	`MyBrand`, `mybrand`, `MYBRAND`
`^(?!Test).+`	أي عنوان غير فارغ لا يبدأ بـ "Test"	`MyBrand` (مسموح)، `TestBrand` (مرفوض)

إدخالات متعددة

عندما يتم تكوين إدخالات متعددة، يُسمح بالعنوان المصدر إذا تطابق مع **أي** إدخال في قائمة المسموح بها.

مثال على قائمة المسموح بها: `^MyBrand$, ^614, ^\+61400000001$`

عنوان المصدر	المطابقات	النتيجة
`MyBrand`	`^MyBrand$`	مسموح
`61412345678`	`^614`	مسموح
`+61400000001`	`^\+61400000001$`	مسموح
`OtherBrand`	لا شيء	مرفوض
`61500000001`	لا شيء	مرفوض

معالجة الأخطاء

عندما يتم رفض `submit_sm` بـ بسبب انتهاك قائمة العناوين المسموح بها، تستجيب البوابة:

الحقل	القيمة
PDU	`submit_sm_resp`
حالة الأمر	`0x0000000A`
اسم الخطأ	`ESME_RINVSRCADR` (عنوان مصدر غير صالح)
معرف الرسالة	فارغ

يتم تسجيل تحذير مع عنوان المصدر المرفوض واسم النظير:

```
خادم SMPP: تم رفض submit_sm من partner_acme - source_addr
'UnauthorisedBrand' غير موجود في قائمة المسموح بها
```

تُعتبر أنماط regex غير الصالحة غير مطابقة (يتم رفض العنوان) ويتم تسجيل تحذير.

التكوين

عبر واجهة الويب

1. انتقل إلى **SMPP > نظائر الخادم**
2. انقر على **تحرير** على النظير المستهدف (أو **إضافة نظير خادم جديد**)
3. ابحث عن حقل **قائمة العناوين المسموح بها** (المسموح بها IP أسفل قائمة)
4. أدخل أنماط regex مفصولة بفواصل:

   ```
   ^MyBrand$,^614,^\+61400000001$
   ```

5. انقر على **حفظ**

تدخل التغييرات حيز التنفيذ على الفور للـ `submit_sm` PDUs الجديدة على الاتصالات الحالية.

عبر ملف التكوين

أضف `source_address_whitelist` إلى تكوين ربط الخادم في `runtime.exs`:

```
config :omnimessage_smpp, :server_binds, [
  %{
    name: "partner_acme",
    system_id: "acme_corp",
    password: "secure_password",
    allowed_bind_types: [:transmitter, :receiver, :transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    source_address_whitelist: ["^MyBrand$", "^614",
"^\+61400000001$"],
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]
```

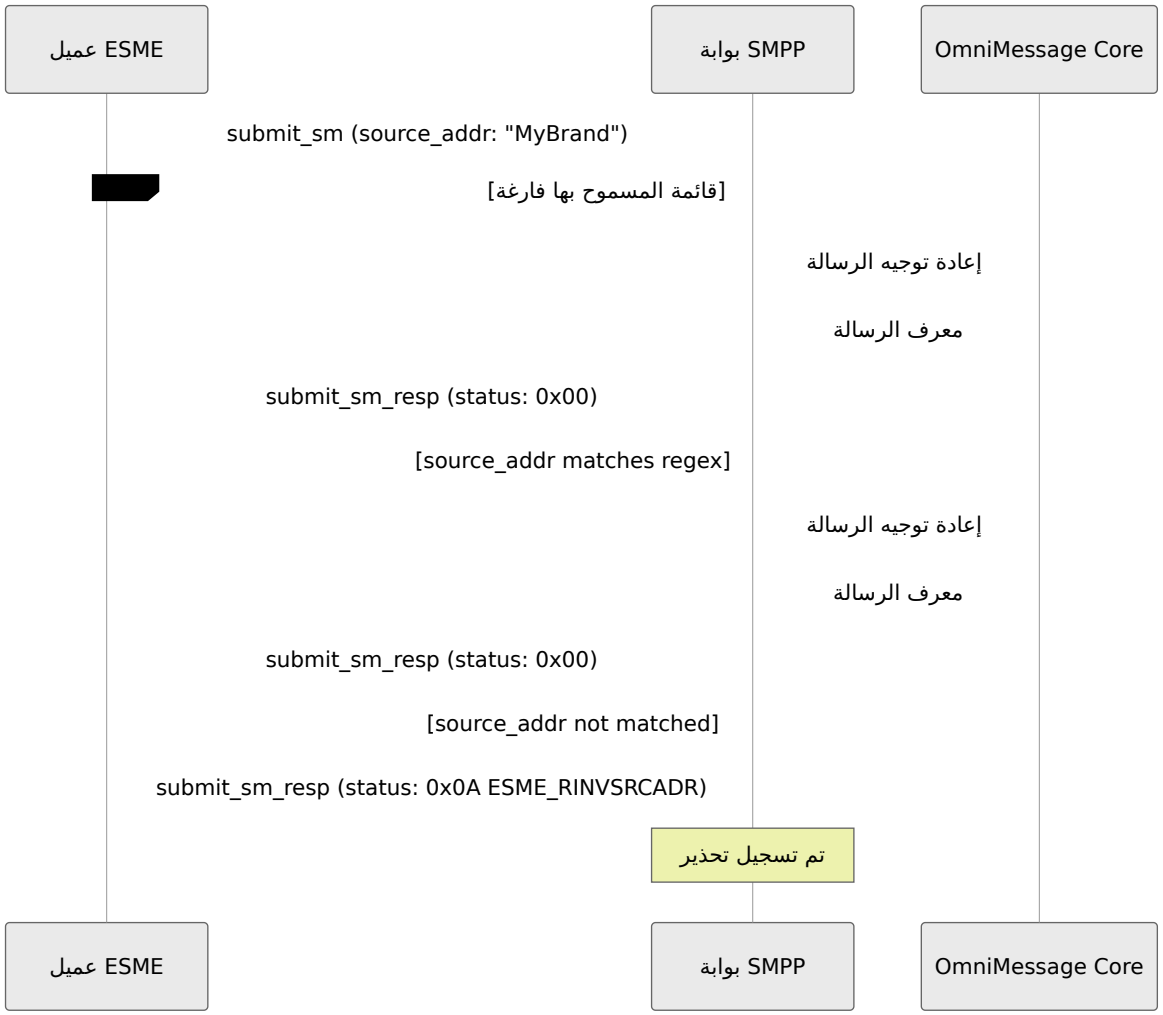
المعلمات

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`source_address_whitelist`	قائمة من السلاسل	لا	`[]` (يسمح للجميع)	قائمة أنماط regex المسموح بها للعناوين المصدرية. القائمة الف��رغة تسمح بجميع العناوين المصدرية.

الترحيل

يتم ترحيل نظائر الخادم الموجودة تلقائيًا عند بدء البوابة. النظائر التي تم إنشاؤها قبل إضافة هذه الميزة تتلقى قائمة فارغة (جميع العناوين المصدرية مسموح بها)، مما يحافظ على السلوك الحالي.

تدفق التحقق



أمثلة

تقييد إلى علامة تجارية واحدة

السماح فقط بالرسائ�من معرف المرسل `AcmeCorp`:

```
^AcmeCorp$
```

السماح بنطاق أرقام أسترالية

السماح بأي رقم هاتف محمول أسترالي (يبدأ بـ `614`):

```
^614
```

السماح بنطاقات أرقام محددة

السماح بأرقام الهواتف المحمولة الأسترالية 614 و 615 (بالضبط 11 رقمًا):

```
^61[45]\d{8}$
```

مزيج من الأحرف الأبجدية الرقمية والأرقام

السماح باسم علامة تجارية ونطاق من الأرقام:

```
^AcmeCorp$,^614,^\+61290000001$
```

مطابقة علامة تجارية غير حساسة لحالة الأحرف

السماح باسم علامة تجارية بغض النظر عن الحالة:

```
(?i)^acmecorp$
```

السماح بالجميع (افتراضي)

اترك الحقل فارغًا للسماح بأي عنوان مصدر. هذا هو السلوك الافتراضي.

استكشاف الأخطاء وإصلاحها

الرسائل المرفوضة مع ESME_RINVSRCADR

الأعراض: الشريك يبلغ عن `submit_sm_resp` مع حالة الأمر `0` و`x0A`.

الأسباب المحتملة:

- في قائمة المسموح بها regex عنوان المصدر لا يتطابق مع أي نمط
- يحتوي على خطأ في بناء الجملة (يُعتبر غير مطابق) نمط regex
- عدم وجود مثبتات (`^`/`$`) مما يتسبب في مطابقة جزئية غير متوقعة أو رفض
- المطابقة حساسة لحالة الأحرف بشكل افتراضي؛ استخدم `(?i)` عدم تطابق الحالة (لمطابقة غير حساسة لحالة الأحرف)

الحل:

1. تحقق من قائمة العناوين المسموح بها للنظير الخادم في واجهة الويب
2. قارن عنوان المصدر المرفوض مع كل نمط regex
3. اختبر الأنماط باستخدام مختبر regex في Elixir: `Regex.match? (~r/^pattern$/, "address")`
4. أضف النمط المفقود أو اضبط الأنماط الموجودة
5. تذكر الهروب من الأحرف الخاصة في regex (الحرفي) مثل `+\` لـ `+`

قائمة المسموح بها لا تأخذ التأثير

الأعراض: الرسائل مقبولة على الرغم من عدم تطابق عنوان المصدر مع قائمة المسموح بها.

الأسباب المحتملة:

- قائمة المسموح بها فارغة (تسمح بالجميع بشكل افتراضي)
- النمط غير مثبت ويتطابق كجزء فرعي (مع `614` يتطابق مثل `X614Y`)
- متصل بنظير خادم مختلف ESME
- تغيير ملف التكوين لم يتم تطبيقه بعد (يتطلب إعادة تشغيل)

الحل:

1. تحقق من أن قائمة المسموح بها ممتلئة (ليست فارغة) في واجهة الويب
2. تأكد من أن الأنماط تستخدم مثبتات `^` و `$` حيثما كانت المطابقة الدقيقة مقصودة
3. تحقق من أي نظير خادم يرتبط به ESME في الحالة الحية
4. إذا كنت تستخدم ملف التكوين، أعد تشغيل الخدمة

الوثائق ذات الصلة

- الوثائق الكاملة لبارامترات نظير الخادم - **مرجع التكوين**
- إدارة اتصالات SMPP - **دليل الاستخدام**
- إجراءات استكشاف الأخطاء وإصلاحها العامة - **استكشاف الأخطاء وإصلاحها**

دليل استكشاف الأخطاء وإصلاحها

المشاكل والحلول الشائعة

الخدمة لا تبدأ / تعارضات المنافذ

تعطل بوابة OmniMessage Core مع SMPP أو على المنفذ 8443 `:eaddrinuse`

الأعراض:

- يتم إعادة تشغيل إحدى الخدمات باستمرار (`systemctl show <unit> -p NRestarts` يرتفع بسرعة).
- تظهر السجلات:

```
Running ApiEx.Endpoint with Bandit ... at https failed,
port 8443 already in use
... failed to start child: ApiEx.Supervisor ... ** (EXIT)
:eaddrinuse
```

- `curl -sk https://<host>:8443/api/messages` يعيد `404` على الرغم من أن المتجر يجب أن يجيب.

السبب: بوابة OmniMessage Core و SMPP موجودتان على نفس المضيف وكلاهما يربطان على المنفذ الافتراضي `8443`. الذي يبدأ أولاً يفوز بالمنفذ؛ الآخر `api_ex` واجهة برمجة التطبيقات من قبل البوابة (التي ليس `/api/messages` يدخل في حلقة تعطل. إذا فازت البوابة، يتم تقديم 404 لديها متجر) ويعيد.

الحل: نقل واجهة برمجة التطبيقات Core الخاصة بالبوابة من منفذ OAM/Management وإلغاء تعليق `/etc/omnimessage_smpp/runtime.exs` قم بتحرير. (بـ `8443` Core يحتفظ) مع تعيين منفذ مجاني. تظل البوابة عميلًا لواجهة برمجة تطبيقات المتجر الخاصة بـ `api_ex`، تجاوز Core عبر `api_base_url`.

```
# /etc/omnimessage_smpp/runtime.exs
config :api_ex,
  api:
    Application.get_env(:api_ex, :api, %{})
    |> Map.put(:port, 8444)
    |> Map.put(:listen_ip, "0.0.0.0")
```

ثم أعد تشغيل كلا الخدمتين:

```
sudo systemctl restart omnimessage-smpp
sudo systemctl restart omnimessage        # let Core reclaim :8443
```

بعد إعادة التشغيل، تحقق من أن Core يمتلك 8443 وأن البوابة تمتلك منفذها الجديد:

```
ss -lntp | grep -E ':8443|:8444'
curl -sk -o /dev/null -w '%{http_code}\n'
https://127.0.0.1:8443/api/messages   # store answers (not 404)
```

انظر CONFIGURATION.md → OAM/Management API port للحصول على التفاصيل.

مشاكل الاتصال بـ OmniMessage

فإن مشاكل الاتصال بـ OmniMessage Core، بلا حالة وتعتمد بالكامل على SMPP نظرًا لأن بوابة OmniMessage هي أكثر المشاكل حرجة.

أعراض قطع الاتصال بـ OmniMessage

- **لا توجد رسائل صادرة**: تتراكم الرسائل، ولا يتم إرسال الرسائل
- **لا توجد رسائل واردة**: لا يمكن للشركاء تقديم الرسائل
- **أوقات الانتظار**: تستغرق مكالمات API وقتًا طويلاً أو تتعطل
- **تظهر السجلات**: "تم رفض الاتصال"، "انتهت المهلة"، "HTTP 503"، "إعادة تعيين الاتصال"

التشخيص

1. تحقق من توفر OmniMessage:

```
# اختبار الاتصال
curl -k -v https://omnimessage-
core.example.com:8443/api/system/health

# اختبار من مضيف البوابة بشكل محدد
ssh gateway-server 'curl -k https://omnimessage-
core.example.com:8443/api/system/health'
```

2. تحقق من عنوان URL لواجهة برمجة التطبيقات المكونة:

```
# مراجعة التكوين
grep -A1 'api_base_url' /opt/omnimessage-smpp/config/runtime.exs

# تحقق من الاتصال بالشبكة
ping omnimessage-core.example.com
nc -zv omnimessage-core.example.com 8443
```

3. تحقق من سجلات البوابة لأخطاء واجهة برمجة التطبيقات:

```
# ابحث عن الأخطاء المتعلقة بواجهة برمجة التطبيقات
sudo journalctl -u omnimessage-smpp -f | grep -i
'api\|omnimessage\|connect'

# ابحث في السجلات عن الأخطاء الأخيرة
sudo journalctl -u omnimessage-smpp -n 200 | grep -i error
```

الحلول

إذا كانت OmniMessage معطلة:

1. اتصل بفريق عمليات OmniMessage
2. ستتراكم الرسائل المعلقة في قائمة الانتظار
3. ستستمر البوابة في إعادة المحاولة (انظر `SMPP_POLL_INTERVAL`)

4. تحقق من صفحة حالة OmniMessage أو المراقبة

إذا كانت OmniMessage تعمل ولكن البوابة لا تستطيع الوصول إليها:

1. تحقق من قواعد جدار الحماية للسماح بـ HTTPS الصادرة
2. تحقق من حل DNS: `nslookup omnimessage-core.example.com`
3. تحقق من توجيه الشبكة: `traceroute omnimessage-core.example.com`
4. إذا كنت تستخدم HTTPS تحقق من شهادات SSL

إذا كان عنوان URL لواجهة برمجة التطبيقات غير مكون بشكل صحيح:

1. قم بتحرير `/opt/omnimessage-smpp/config/runtime.exs`
2. تحقق من أن `api_base_url` صحيح (يجب أن يكون HTTPS للإنتاج)
3. أعد تشغيل البوابة: `sudo systemctl restart omnimessage-smpp`

مشاكل الاتصال

لا يمكن إنشاء الاتصال

الأعراض:

- الحالة تظهر "غير متصل" (أحمر)
- لا يوجد ربط ناجح في السجلات
- محاولات اتصال متكررة

الأسباب المحتملة والحلول:

1. مشاكل الاتصال بالشبكة

تحقق:

```
# اختبار حل DNS
nslookup smpp.carrier.com

# اختبار الاتصال
ping -c 3 smpp.carrier.com

# اختبار المنفذ
telnet smpp.carrier.com 2775
# أو
nc -zv smpp.carrier.com 2775
```

الحلول:

- إذا فشل DNS: استخدم عنوان IP بدلاً من اسم المضيف في التكوين
- إذا فشل ping: تحقق من قواعد جدار الحماية، اتصل بالناقل
- إذا فشل المنفذ: تحقق من رقم المنفذ الصحيح، تحقق من جدار الحماية

2. بيانات اعتماد غير صحيحة

تحقق:

- تظهر السجلات "فشل الربط" أو "خطأ في المصادقة"
- واجهة الويب: SMPP → تحقق من أقران العملاء system_id وكلمة المرور

الحلول:

- تأكيد بيانات الاعتماد مع الناقل
- تحقق من الأخطاء المطبعية (حساسة لحالة الأحرف)
- تحديث التكوين وإعادة الاتصال

3. عنوان IP غير مدرج في القائمة البيضاء

تحقق:

- تم رفض الاتصال على الفور
- تظهر سجلات الناقل عنوان IP غير المصرح به

الحلول:

- تأكيد عنوان IP العام للبوابة الخاصة بك:

```
curl ifconfig.me
```

- طلب من الناقل إضافة IP إلى القائمة البيضاء
- تحقق من أن IP لم يتغير (IP ديناميكي)

4. جدار الحماية يحظر

تحقق:

```
# تحقق مما إذا كان المنفذ مفتوحًا
sudo iptables -L -n | grep 2775

# تحقق من UFW (أوبونتو/ديبيان)
sudo ufw status | grep 2775

# تحقق من firewalld (RHEL/CentOS)
sudo firewall-cmd --list-ports | grep 2775
```

الحلول:

```
# أوبونتو/ديبيان
sudo ufw allow out 2775/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=2775/tcp
sudo firewall-cmd --reload
```

5. عنوان/منفذ المصدر المثبت غير متاح

ينطبق فقط عندما يحدد عميل الربط `bind_source_ip` و/أو `bind_source_port` (انظر CONFIGURATION.md).

تحقق:

- تظهر السجلات فشل الاتصال في فتح المقبس المحلي
- `bind_source_ip` ليس عنوانًا تحتفظ به هذه المضيف
- `bind_source_port` المكون يحتفظ عملية أخرى بالفعل بـ

الحلول:

- تأكيد أن `bind_source_ip` هو عنوان محلي: `ip addr show`
- غير صالح ويتم تجاهله - ثم يختار نظام التشغيل عنوان `bind_source_ip` يتم تسجيل المصدر، والذي قد لا يتطابق مع القائمة البيضاء للناقل. صحح العنوان إذا كانت التسليم تفشل.
- لذا يعيد الاتصال الربط على الفور، `SO_REUSEADDR` يتم فتح المنفذ المثبت باستخدام إذا كان المنفذ لا يزال غير متاح، `TIME_WAIT` حتى أثناء وجود المقبس القديم في تحقق مما إذا كانت خدمة أخرى تستخدمه: `sudo ss -ltnp | grep <port>`

الاتصال يستمر في الانقطاع

الأعراض:

- تم إنشاء الاتصال ولكن يتم قطع الاتصال بشكل متكرر
- زيادة مقياس `smpp_disconnection_total`
- تظهر السجلات إعادة اتصالات متكررة

الأسباب المحتملة والحلول:

1. عدم استقرار الشبكة

تحقق:

```
# مراقبة فقدان الحزم
ping -c 100 smpp.carrier.com | grep loss

# تحقق من أخطاء الشبكة
netstat -s | grep -i error
```

الحلول:

- اتصل بالناقل حول مشاكل الشبكة
- تحقق مع مزود خدمة الإنترنت إذا كان من جانبك
- النظر في اتصال/مسار احتياطي

2. انتهاء مهلة استفسار الرابط

تحقق:

- تظهر السجلات "انتهاء مهلة استفسار الرابط"
- ينقطع الاتصال بعد فترات من عدم النشاط

الحلول:

- مدة انتهاء المهلة الافتراضية هي 30 ثانية
- تحقق من أن الشبكة تسمح بحزم keepalive
- تحقق من جدران الحماية العدوانية التي تنتهي مهلة الاتصالات الخاملة

3. تجاوز حد TPS

تحقق:

- معدل الرسائل مرتفع عند وقت قطع الاتصال
- يقوم الناقل بتقليل الرسائل

الحلول:

- مراجعة إعداد `tps_limit`
- تقليل TPS للناقل إلى 70-80% من الحد الأقصى
- توزيع حركة المرور عبر عمليات ربط متعددة

4. مشاكل خادم الناقل

تحقق:

- تحقق من حالة خدمة الناقل
- اتصل بدعم الناقل

الحلول:

- انتظر حتى يحل الناقل المشكلة
- تكوين ناقل احتياطي إذا كان متاحًا

مشاكل تسليم الرسائل

الرسائل لا تُرسل

الأعراض:

- الرسائل عالقة في قائمة الانتظار
- `smpp_messages_sent_total` لا يرتفع
- الاتصال يظهر متصل

الأسباب المحتملة والحلول:

1. توجيه `dest_smsc` غير صحيح

تحقق:

- للرسالة `dest_smsc` واجهة الويب ← قائمة الانتظار ← تحقق من حقل
- الحالة الحية ← SMPP قارن مع اسم الاتصال في

الحلول:

- يتم توجيه الرسائل بناءً على حقل `dest_smsc`
- الصحيح `dest_smsc` تحقق من أن الخلفية تعين
- إذا كان `dest_smsc` NULL، تحقق من التوجيه الافتراضي

2. الرسائل المجدولة للمستقبل

تحقق:

- واجهة الويب ← قائمة الانتظار ← تحقق من حقل `deliver_after`
- الرسائل ذات الطابع الزمني المستقبلي لن تُرسل بعد

الشرح:

- للرسائل الفاشلة `deliver_after` يقوم نظام إعادة المحاولة بتعيين
- تنتظر الرسائل حتى ذلك الوقت قبل إعادة المحاولة

الحلول:

- انتظر حتى الوقت المجدول
- إذا كان الأمر عاجلاً، اتصل بفريق الخلفية لإعادة تعيين الطابع الزمني

3. حد TPS منخفض جدًا

تحقق:

- تراكم كبير في قائمة الانتظار
- الرسائل تُرسل ببطء شديد

الحلول:

- زيادة `tps_limit` في التكوين
- تحقق من أن الناقل يمكنه التعامل مع معدل أعلى
- انظر CONFIGURATION.md

4. عامل قائمة الانتظار غير قيد التشغيل

تحقق:

- حالة الخدمة
- السجلات للأخطاء

الحلول:

```
# إعادة تشغيل الخدمة
sudo systemctl restart omnimessage-smpp

# تحقق من السجلات
sudo journalctl -u omnimessage-smpp -f
```

5. الواجهة الأمامية غير مسجلة تحت الاسم المتوقع

يقوم OmniMessage Core بتوجيه الرسالة إلى هذه البوابة فقط إذا تم تسجيل نظير تحت اسم **مفتاح التوجيه** الخاص به `dest_smsc` للرسالة. يسجل كل نظير تحت الواجهة الأمامية يساوي (انظر تسجيل الواجهة الأمامية):

- **أقران العملاء**: قائمة الانتظار المكونة، أو اسم النظير العاري إذا لم يتم تعيين أي `queue`.

- اسم الربط: **عمليات الربط**.

تحقق:

- السجلات للمدخلات `Successfully registered SMPP peer frontend '<name>'` — تأكد من أن `<name>` يتطابق مع `dest_smsc` على الرسائل العالقة
- توقع تسجيلًا واحدًا تحت اسم قائمة الانتظار، `queue` بالنسبة للربط التي تشارك المشتركة، وليس واحدًا لكل نظير

الحلول:

- تأكد من أن الخلفية تعين `dest_smsc` إلى مفتاح التوجيه (قائمة انتظار النظير، وإلا (نظام التسمية السابق الذي لم يعد ينطبق) `{hostname}_{peer}` اسمه)، وليس إلى
- إذا كنت تستخدم `queue` مشتركة، قم بتوجيه الرسائل إلى اسم قائمة الانتظار بدلاً من اسم نظير فردي
- تأكد من أن النظير مفعل — النظائر المعطلة غير مسجلة

معدل فشل التسليم مرتفع

الأعراض:

- `smpp_delivery_failures_total` يرتفع
- تظهر السجلات "submit_sm_resp" مع حالة خطأ
- الرسائل لا تصل إلى المستلمين

الأسباب المحتملة والحلول:

1. أرقام وجهة غير صالحة

تحقق:

- السجلات لأكواد الخطأ المحددة
- مراجعة تنسيق وجهة الرسالة

أكواد الخطأ الشائعة:

- `0x0000000B` - وجهة غير صالحة
- `0x00000001` - طول رسالة غير صالح

- `0x00000003` - أمر غير صالح

الحلول:

- تحقق من تنسيق الرقم (يوصى بـ E.164)
- تحقق من أن الرقم يتضمن رمز الدولة
- تحقق من متطلبات الناقل

2. محتوى الرسالة غير صالح

تحقق:

- طول الرسالة
- الأحرف الخاصة
- الترميز

الحلول:

- GSM-7: الحد الأقصى 160 حرفًا
- UCS-2: الحد الأقصى 70 حرفًا
- إزالة الأحرف غير المدعومة
- تحقق من إعدادات الترميز

3. رفض الناقل

تحقق:

- أكواد الخطأ المحددة من الناقل
- أنماط الرسائل المرفوضة

الحلول:

- اتصل بالناقل لمعرفة سبب الرفض
- قد تحتاج إلى تصفية المحتوى
- تحقق من أنماط البريد العشوائي/الإساءة

4. الرسائل منتهية الصلاحية

تحقق:

- الطابع الزمني `expires` للرسالة
- توقيت محاولة التسليم

الحلول:

- زيادة فترة صلاحية الرسالة
- تقليل تأخير إعادة المحاولة للرسائل الحساسة للوقت

مشاكل واجهة الويب

لا يمكن الوصول إلى لوحة التحكم على الويب

الأعراض:

- المتصفح لا يمكنه الاتصال بـ https://your-server:8087
- انتهاء المهلة أو تم رفض الاتصال

الأسباب المحتملة والحلول:

1. الخدمة غير قيد التشغيل

تحقق:

```
sudo systemctl status omnimessage-smpp
```

الحلول:

```
# إذا كانت متوقفة، ابدأها
sudo systemctl start omnimessage-smpp

# تحقق من السجلات للأخطاء
sudo journalctl -u omnimessage-smpp -n 50
```

2. 8087 جدار الحماية يحظر المنفذ

تحقق:

```
sudo ufw status | grep 8087
# أو
sudo firewall-cmd --list-ports | grep 8087
```

الحلول:

```
# أوبونتو/ديبيان
sudo ufw allow 8087/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=8087/tcp
sudo firewall-cmd --reload
```

3. مشاكل شهادة SSL

تحقق:

- يظهر المتصفح تحذير أمان
- الشهادة منتهية الصلاحية أو غير صالحة

الحلول:

- قبول استثناء الأمان (إذا كانت موقعة ذاتيًا)
- تثبيت شهادة SSL صالحة
- تحقق من وجود ملفات الشهادة:
  ```
  ls -l /opt/omnimessage-smpp/priv/cert/
  ```

4. عنوان URL غير صحيح

تحقق:

- تحقق من استخدام HTTPS (ليس HTTP)
- تحقق من عنوان IP/اسم المضيف الصحيح
- تحقق من المنفذ 8087

واجهة الويب تظهر أخطاء

الأعراض:

- يتم تحميل الصفحة ولكن تظهر أخطاء
- لا تعمل الوظائف
- البيانات لا تعرض

الحلول:

1. **مسح ذاكرة التخزين المؤقت للمتصفح**:
    - Ctrl+F5 (تحديث صعب)
    - مسح ذاكرة التخزين المؤقت للمتصفح وملفات تعريف الارتباط
2. **تحقق من وحدة تحكم المتصفح**:
    - اضغط على F12
    - تحقق من علامة التبويب "وحدة التحكم" لأخطاء JavaScript
    - أبلغ الدعم إذا تم العثور على أخطاء
3. **جرب متصفحًا مختلفًا**:
    - اختبر في Edge وFirefox وChrome
    - عزل المشكلات الخاصة بالمتصفح
4. **تحقق من سجلات الخدمة**:

```
sudo journalctl -u omnimessage-smpp -f
```

مشاكل المقاييس

مقاييس Prometheus غير متاحة

الأعراض:

- `curl http://localhost:4000/metrics` يفشل
- لا يمكن لـ Prometheus جمع المقاييس
- استجابة فارغة أو خطأ

الأسباب المحتملة والحلول:

1. الخدمة غير قيد التشغيل

تحقق:

```
sudo systemctl status omnimessage-smpp
```

الحلول:

```
sudo systemctl start omnimessage-smpp
```

2. المنفذ غير متاح

تحقق:

```
# اختبار محليًا
curl http://localhost:4000/metrics

# اختبار عن بُعد
curl http://your-server-ip:4000/metrics
```

الحلول:

- إذا كانت المحلية تعمل ولكن البعيدة لا تعمل: تحقق من جدار الحماية
- افتح المنفذ 4000 في جدار الحماية لخادم Prometheus

3. نقطة النهاية غير صحيحة

تحقق:

- نقطة النهاية هي `/metrics` (ليس `/prometheus` أو `/stats`)
- المنفذ هو 4000 (ليس 8087)

المقاييس تظهر قيم غير متوقعة

الأعراض:

- تعود العدادات إلى الصفر
- تظهر المقاييس قيم خاطئة
- المقاييس مفقودة لبعض الربط

الحلول:

1. **إعادة تشغيل الخدمة تعيد تعيين العدادات**:
    - يتم إعادة تعيين العدادات عند إعادة تشغيل الخدمة
    - هذا سلوك طبيعي
    - استخدم `increase()` أو `rate()` في استعلامات Prometheus
2. **الربط الجديدة لا تظهر**:
    - تظهر المقاييس فقط بعد الحدث الأول
    - أرسل رسالة اختبار لتعبئة المقاييس
    - تحقق من أن الربط مفعل ومتصل
3. **مقاييس قديمة**:
    - قد تظهر الربط القديمة في المقاييس
    - أعد تشغيل الخدمة لمسح الإدخالات القديمة
    - للتصفية Prometheus أو استخدم إعادة تسمية

مشاكل الأداء

استخدام CPU مرتفع

تحقق:

```
top -p $(pgrep -f omnimessage-smpp)
```

الأسباب المحتملة:

- حجم رسالة مرتفع جدًا
- عدد كبير من الاتصالات
- مشكلة في التكوين

الحلول:

- تحقق من أن معدل الرسائل ضمن السعة
- مراجعة حدود TPS
- اتصل بالدعم إذا كان استخدام CPU مرتفعًا بشكل مستمر

استخدام الذاكرة مرتفع

تحقق:

```
ps aux | grep omnimessage-smpp
```

الأسباب المحتملة:

- قائمة انتظار الرسائل الكبيرة في الذاكرة
- تسرب الذاكرة (نادراً)

الحلول:

- أعد تشغيل الخدمة لمسح الذاكرة
- تحقق من حجم قائمة انتظار الرسائل
- اتصل بالدعم إذا استمرت الذاكرة في النمو

معالجة الرسائل ببطء

الأعراض:

- تستغرق الرسائل وقتًا طويلًا للإرسال

- تتراكم قائمة الانتظار
- معدل الرسائل منخفض

تحقق:

1. ودد TPS - قد تكون صارمة جدًا
2. `queue_check_frequency` - قد تكون مرتفعة جدًا
3. وقت استجابة واجهة برمجة التطبيقات الخلفية - قد تكون بطيئة
4. زمن الانتقال الشبكي إلى الناقل

الحلول:

- إذا كان الناقل يسمح بذلك TPS زيادة
- تقليل `queue_check_frequency` لزيادة سرعة الاستطلاع
- تحسين واجهة برمجة التطبيقات الخلفية
- تحقق من زمن الانتقال الشبكي

مشاكل التكوين

أخطاء في بناء جملة ملف التكوين

الأعراض:

- الخدمة لا تبدأ بعد تغيير التكوين
- تظهر السجلات "خطأ في بناء الجملة" أو "خطأ في التحليل"

تحقق:

```
# تحقق من بناء جملة Elixir
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

الأخطاء الشائعة:

- فاصلة مفقودة بين إدخالات الخريطة

- اقتباسات غير متطابقة (" مقابل ')
- أقواس أو أقواس غير متطابقة
- مفقود `import Config` في الأعلى

الحلول:

- استعادة من النسخة الاحتياطية
- مراجعة بناء الجملة بعناية
- استخدم محرر نصوص مع تمييز بناء جملة Elixir

التغييرات لا تؤثر

الأعراض:

- تم تعديل التكوين ولكن لا يوجد تغيير في السلوك
- لا تزال الإعدادات القديمة نشطة

الحلول:

```
# تتطلب تغييرات التكوين إعادة التشغيل
sudo systemctl restart omnimessage-smpp

# تحقق من نجاح إعادة التشغيل
sudo systemctl status omnimessage-smpp

# تحقق من السجلات للأخطاء
sudo journalctl -u omnimessage-smpp -n 50
```

استعادة الطوارئ

فشل النظام بالكامل

الخطوات:

1. **تحقق من صحة النظام الأساسية**:

```
# مساحة القرص
df -h

# الذاكرة
free -h

# تحميل CPU
uptime
```

2. **تحقق من حالة الخدمة**:

```
sudo systemctl status omnimessage-smpp
```

3. **راجع السجلات الأخيرة**:

```
sudo journalctl -u omnimessage-smpp -n 200
```

4. **حاول إعادة تشغيل الخدمة**:

```
sudo systemctl restart omnimessage-smpp
```

5. **إذا فشلت إعادة التشغيل**:

   - تحقق من بناء جملة التكوين
   - تحقق من وجود شهادات SSL
   - تحقق من أذونات الملفات
   - راجع السجلات للخطأ المحدد

6. (إذا لزم الأمر) **استعادة من النسخة الاحتياطية**:

```
# استعادة التكوين
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup \
  /opt/omnimessage-smpp/config/runtime.exs

# إعادة التشغيل
sudo systemctl restart omnimessage-smpp
```

7. **اتصل بالدعم** إذا لم يتم حل المشكلة

الحصول على المساعدة

المعلومات التي يجب جمعها

قبل الاتصال بالدعم، اجمع:

1. **الإصدار**: `cat /opt/omnimessage-smpp/VERSION`
2. **السجلات الأخيرة**:
   ```
   sudo journalctl -u omnimessage-smpp -n 200 > /tmp/smpp-logs.txt
   ```
3. **التكوين** (قم بتطهير كلمات المرور):
   ```
   sudo cp /opt/omnimessage-smpp/config/runtime.exs /tmp/config.exs
   # تحرير /tmp/config.exs لإزالة كلمات المرور قبل الإرسال
   ```
4. **مخرجات المقاييس**:
   ```
   curl http://localhost:4000/metrics > /tmp/metrics.txt
   ```
5. **معلومات النظام**:
   ```
   uname -a > /tmp/system-info.txt
   free -h >> /tmp/system-info.txt
   df -h >> /tmp/system-info.txt
   ```

الاتصال بالدعم

- **البريد الإلكتروني**: support@omnitouch.com
- **الهاتف**: +61 XXXX XXXX (24/7)
- **تضمين**: جميع المعلومات من أعلاه

الوثائق ذات الصلة

- **USAGE.md** - إجراءات التشغيل العادية
- **CONFIGURATION.md** - مرجع التكوين
- **MONITORING.md** - المراقبة والمقاييس
- **README.md** - نظرة عامة على النظام

دليل العمليات

إجراءات التشغيل اليومية

الاعتماد الحرج: OmniMessage Core

مهم: لا يمكن لجسر OmniMessage SMPP العمل بدون الوصول إلى OmniMessage Core. الجسر هو مجرد مترجم بروتوكول - OmniMessage تتم جميع معالجة الرسائل في.

إذا أصبح OmniMessage غير متاح:

- ❌ لا يمكن تقديم رسائل جديدة
- ❌ لا يمكن استرجاع الرسائل المعلقة
- ❌ لا يمكن الإبلاغ عن حالة التسليم
- ❌ يبدو أن النظام يتجمد أو ينتهي وقته

تحقق من صحة OmniMessage:

```
# اختبار اتصال API
curl -k https://omnimessage-
core.example.com:8443/api/system/health

# تحقق من عنوان URL الخاص بـ API في السجلات
grep api_base_url /opt/omnimessage-smpp/config/runtime.exs
```

العمليات اليومية

فحص الصحة الصباحية

قم بإجراء هذه الفحوصا❓❓ في بداية كل يوم:

1. **الوصول إلى لوحة التحكم على الويب**
   - URL: `https://your-server:8087`

- تحقق مما إذا كانت لوحة التحكم تُحمّل بشكل صحيح

2. **تحقق من حالة الاتصال**

   - انتقل إلى: SMPP → الحالة المباشرة
   - تحقق من أن جميع الاتصالات تظهر "متصلة" (باللون الأخضر)
   - لاحظ أي روابط مفصولة

3. **مراجعة مقاييس الرسائل**

   - انتقل إلى: علامة التبويب قائمة الانتظار
   - تحقق من أن عدد الرسائل معقول
   - تحقق من عدم وجود تراكم غير متوقع في قائمة الانتظار

4. **تحقق من سجلات النظام**

   - انتقل إلى: علامة التبويب السجلات
   - ابحث عن رسائل الخطأ (باللون الأحمر)
   - لاحظ أي أنماط تحذيرية

5. **مراجعة مقاييس Prometheus**

   - `curl http://localhost:4000/metrics`
   - أو تحقق من لوحات Grafana
   - تحقق من أن معدلات الرسائل طبيعية

المراقبة المستمرة

قم بإعداد تنبيهات لـ:

- فشل الاتصال (> 2 دقيقة غير متصلة)
- معدلات فشل التسليم العالية (> 5%)
- عدم وجود حركة مرور لفترات طويلة
- انقطاعات متكررة

انظر MONITORING.md لتكوين التنبيهات.

فهم توجيه الرسائل

يقوم الجسر بتوجيه الرسائل بين OmniMessage Core واتصالات SMPP باستخدام حقلين رئيسيين:

- `dest_smsc` - عندما يقوم **روابط العملاء**. يوجه الرسائل الصادرة إلى OmniMessage بتجميع رسالة مع `dest_smsc: "vodafone_uk"`، تلتقط رابط العميل المسماة `vodafone_uk` وترسلها عبر SMPP `submit_sm` الرسالة.
- `source_smsc` - عندما يقوم **روابط الخادم**. يوجه الرسائل الواردة إلى OmniMessage بتجميع رسالة مع `source_smsc: "partner_acme"`، يقوم الجسر بتسليمها إلى العملاء المتصلين برابط الخادم المسماة `partner_acme` عبر SMPP `deliver_sm`.

تمييز رئيسي: روابط العملاء ترسل PDUs من نوع `submit_sm` (الجسر هو ESME يقدم إلى SMSC الذي يسلم إلى الناقل). روابط الخادم ترسل PDUs من نوع `deliver_sm` (الجسر هو متصل ESME).

تسجيل الواجهة الأمامية

يقوم الجسر بتسجيل نفسه تلقائيًا مع OmniMessage Core حتى يعرف الجزء الخلفي أي اتصالات SMPP متاحة لتوجيه الرسائل.

- **اسم التسجيل**: يتم التحكم فيه بواسطة مفتاح التكوين `smsc_name` (افتراضي: `"smpp_gateway"`، env: `SMSC_NAME`)
- **نبض القلب**: يُرسل كل 60 ثانية للحفاظ على التسجيل نشطًا
- **انتهاء الصلاحية**: تنتهي صلاحية التسجيل في الجزء الخلفي بعد 90 ثانية بدون نبض قلب
- **تسجيل لكل رابط**: يتم تسجيل كل نظير مفعل بشكل فردي تحت **مفتاح التوجيه** الخاص به، بحيث يساوي اسم الواجهة الأمامية المسجل `dest_smsc` الذي يتوجه عليه الجزء الخلفي:
  - **الأقران العملاء**: يتم تسجيلها تحت `queue` المكونة لها (إذا تم تعيينها)، وإلا تحت اسم النظير العاري. لذلك، فإن عدة أقران عملاء تشارك واحدة `queue` تسجل تحت ذلك الاسم المشترك الوحيد.
  - **روابط الخادم**: يتم تسجيلها تحت اسم الربط الخاص بها.

إذا توقف الجسر أو فقد الاتصال بـ OmniMessage Core، فإن تسجيلاته تنتهي ويوقف الجزء الخلفي توجيه الرسائل إليه.

استكشاف الأخطاء: إذا لم يتم توجيه الرسائل إلى الجسر، تحقق من:

1. السجلات للمدخلات "frontend_register"
2. تطابق `smsc_name` مع ما يتوقعه OmniMessage
3. الاتصال الشبكي بـ OmniMessage Core (`api_base_url`)

إدارة اتصالات SMPP

كيفية تكوين أقران SMPP

يمكن تكوين اتصالات SMPP (الأقران) **طريقتين** باستخدام:

الطريقة 1: واجهة الويب أو واجهة برمجة التطبيقات OAM (موصى بها)

- **الميزة**: التغييرات تدخل حيز التنفيذ على الفور، لا حاجة لإعادة التشغيل
- **الموقع**: SMPP → علامات تبويب الأقران العملاء / الأقران الخادم، أو على واجهة برمجة `/api/smpp/server_peers` و `/api/smpp/client_peers` التطبيقات OAM
- **العمليات**: إضافة، تحرير، حذف الأقران

- **ما يفعله الحفظ**:
  - **نظير الخادم**: يبدأ عامل قائمة الانتظار الخاص به، أو يعاد تشغيله مع إذا كان هذا هو أول نظير خادم. تبقى SMPP الإعدادات الجديدة. يبدأ مستمع الجلسات المرتبطة قيد التشغيل.
  - **نظير العميل**: يتم فصل الاتصال وإعادة إنشائه مع الإعدادات الجديدة.
  - **تعطيل أو حذف نظير**: يتم إيقاف عماله. يتم فصل نظير العميل أولاً. يتم إرسال `unbind` لجلسات نظير الخادم المرتبطة (تغلق بعد 2 ثانية إذا لم يستجب ESME)، ويتم رفض الروابط الجديدة.
- **الاستمرارية**: مخزنة في قاعدة بيانات Mnesia
- **الأفضل لـ**: العمليات اليومية، الاختبار، التغييرات السريعة

الطريقة 2: ملف التكوين

- **الميزة**: التكوين ككود، التحكم في الإصدار
- **الموقع**: `/opt/omnimessage-smpp/config/runtime.exs`
- **العمليات**: تعريف الأقران في تكوين Elixir
- **الاستمرارية**: تعتمد على الملفات، وتبقى بعد إعادة التشغيل
- **يتطلب**: إع��دة تشغيل الخدمة بعد التغييرات
- **الأفضل لـ**: الإعداد الأولي، البنية التحتية ككود

ملاحظة: يتم تخزين تغييرات واجهة الويب بشكل منفصل وتجاوز إعدادات ملف التكوين.

انظر CONFIGURATION.md للرجوع إلى ملف التكوين.

إضافة اتصال عميل جديد

الغرض ESME (عميل) متصل بـ **SMSC** (خادم) للناقل: تكوين الجسر ليعمل كـ

التحضير: جمع المعلومات من الناقل:

- اسم مضيف خادم SMPP / IP
- رقم المنفذ (عادةً 2775)
- معرف النظام (اسم المستخدم)
- كلمة المرور
- نوع الربط (عادةً ثنائي الاتجاه)
- حد TPS

اختر واحدة من الطرق التالية:

الخيار A: عبر واجهة الويب (موصى بها)

المزايا: تأثير فوري، لا حاجة لإعادة التشغيل

الخطوات:

1. **انتقل إلى الأقران العملاء**:
   - افتح واجهة الويب: `https://your-server:8087`
   - انتقل إلى: SMPP → الأقران العملاء
2. **إضافة نظير جديد**:
   - انقر على "إضافة نظير عميل جديد"
   - املأ النموذج:
     - **الاسم**: `vodafone_uk` (معرف فريد)
     - **المضيف**: `smpp.vodafone.co.uk`
     - **المنفذ**: `2775`
     - **معرف النظام**: `your_username`
     - **كلمة المرور**: `your_password`
     - **نوع الربط**: `Transceiver`
     - **حد TPS**: `100`
     - **تكرار فحص قائمة الانتظار**: `1000`
   - انقر على "حفظ"

3. **تأسيس الاتصال تلقائيًا**:
    - يحاول الجسر الاتصال على الفور
    - انتقل إلى: SMPP → الحالة المباشرة
    - يجب أن يتغير الوضع إلى "متصل" (باللون الأخضر) خلال 10-30 ثانية
    - تحقق من علامة التبويب السجلات للحصول على رسالة الربط الناجحة
4. **اختبار تدفق الرسائل**:
    - انتقل إلى: علامة التبويب قائمة الانتظار
    - قدم رسالة اختبار مع `dest_smsc` تتطابق مع اسم الربط
    - راقب في الحالة المباشرة للإرسال
    - تحقق من تأكيد التسليم

الخيار B: عبر ملف التكوين

المزايا: البنية التحتية ككود، التحكم في الإصدار

الخطوات:

1. **تحرير ملف التكوين**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **إضافة الربط الجديد إلى التكوين**:

```
config :omnimessage_smpp, :binds, [
  # الروابط الموجودة...

  # إضافة الربط الجديد
  %{
    name: "vodafone_uk",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.vodafone.co.uk",
    port: 2775,
    system_id: "your_username",
    password: "your_password",
    tps_limit: 100,
    queue_check_frequency: 1000,

    # اختياري: تثبيت عنوان IP/المنفذ المحلي إذا كان الناقل يدرجهم في القائمة البيضاء
    bind_source_ip: "10.20.30.40",
    bind_source_port: 5000,

    # اختياري: مشاركة قائمة انتظار واحدة في الخلفية عبر عدة روابط (تكون افتراضية على الاسم)
    queue: "vodafone_uk"
  }
]
```

انظر CONFIGURATION.md للحصول على القائمة الكاملة لمعلمات ربط العميل، بما في ذلك `bind_source_ip` و `bind_source_port` `queue`.

3. **حفظ وإعادة تشغيل الخدمة**:

```
# حفظ الملف (Ctrl+X، Y، Enter في nano)

# إعادة تشغيل الخدمة
sudo systemctl restart omnimessage-smpp
```

4. **تحقق من الاتصال**:
   - انتقل إلى: SMPP → الحالة المباشرة
   - ابحث عن الاتصال الجديد

- يجب أن تكون الحالة "متصلة" (باللون الأخضر)
- تحقق من السجلات للحصول على ربط ناجح

5. **اختبار تدفق الرسائل**:

   - انتقل إلى: علامة التبويب قائمة الانتظار
   - تتطابق مع اسم الربط الجديد `dest_smsc` قدم رسالة اختبار مع
   - راقب في الحالة المباشرة للإرسال
   - تحقق من تأكيد التسليم

إضافة ربط خادم

الغرض SMSC يقبل الاتصالات من (خادم) **ESMEs** (عملاء الشركاء): تكوين الجسر ليعمل كـ الخارجيين

التحضير:

1. **إنشاء بيانات الاعتماد**:

   - أنشئ معرف نظام فريد: `partner_name`
   - أنشئ كلمة مرور قوية
   - وثق وشارك بأمان مع الشريك

2. **الحصول على معلومات الشريك**:

   - المصدر للشريك IP عناوين
   - حجم الرسائل المتوقع (لحد TPS)
   - أنواع الربط المطلوبة

اختر واحدة من الطرق التالية:

عبر واجهة الويب (موصى بها) :A الخيار

المزايا: تأثير فوري، لا حاجة لإعادة التشغيل

الخطوات:

1. **انتقل إلى الأقران الخادم**:

   - افتح واجهة الويب: `https://your-server:8087`

- انتقل إلى: SMPP → الأقران الخادم

2. **إضافة نظير خادم جديد**:

   - انقر على "إضافة نظير خادم جديد"
   - املأ النموذج:
     - **الاسم**: `partner_acme` (معرف فريد)
     - **معرف النظام**: `acme_corp`
     - **كلمة المرور**: `secure_password_123`
     - **أنواع الربط المسموح بها**: اختر الكل (مرسل، مستقبل، ثنائي الاتجاه)
     - **قائمة IP البيضاء**: `203.0.113.0/24` (مفصولة بفواصل لعدة)
     - **حد TPS**: `50`
     - **تكرار فحص قائمة الانتظار**: `1000`
   - انقر على "حفظ"

3. **الجسر جاهز للاتصال**:

   - أصبح نظير الخادم نشطًا الآن وينتظر اتصال الشريك
   - لا حاجة لإعادة التشغيل

4. **مشاركة المعلومات مع الشريك**:

   - عنوان IP للجسر
   - المنفذ: `2775`

- معرف النظام: `acme_corp`
- كلمة المرور: `secure_password_123`
- نوع الربط: كما هو مكون

5. **انتظر اتصال الشريك**:
   - انتقل إلى: SMPP → الحالة المباشرة
   - راقب الاتصال الوارد
   - تحقق من نجاح المصادقة
   - تحقق من تطابق IP مع القائمة البيضاء

الخيار B: عبر ملف التكوين

المزايا: البنية التحتية ككود، التحكم في الإصدار

الخطوات:

1. **تحرير ملف التكوين**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **إضافة ربط خادم وتكوين الاستماع**:

```
# أضف إلى قائمة server_binds
config :omnimessage_smpp, :server_binds, [
  # الروابط الخادمة الموجودة...

  # إضافة ربط خادم جديد
  %{
    name: "partner_acme",
    system_id: "acme_corp",
    password: "secure_password_123",
    allowed_bind_types: [:transmitter, :receiver,
:transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]

# تأكد من وجود تكوين الاستماع (يحتاج لمرة واحدة فقط)
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

3. **حفظ وإعادة تشغيل الخدمة**:

```
sudo systemctl restart omnimessage-smpp
```

4. **مشاركة المعلومات مع الشريك**:

   - عنوان IP للجسر
   - المنفذ: `2775`
   - معرف النظام: `acme_corp`
   - كلمة المرور: `secure_password_123`
   - نوع الربط: كما هو مكون

5. **انتظر اتصال الشريك**:

   - انتقل إلى: SMPP → الحالة المباشرة

- راقب الاتصال الوارد
- تحقق من نجاح المصادقة
- تحقق من تطابق IP مع القائمة البيضاء

تعديل اتصال موجود

الغرض: تحديث معلمات الاتصال (حدود TPS، قائمة كلمات المرور، إلخ IP البيضاء.)

اختر واحدة من الطرق التالية:

الخيار A: عبر واجهة الويب (موصى بها)

المزايا: تأثير فوري، لا حاجة لإعادة التشغيل

الخطوات:

1. **انتقل إلى الأقران**:
   - افتح واجهة الويب: `https://your-server:8087`
   - للاتصالات العميلة: SMPP → الأقران العملاء
   - للاتصالات الخادمة: SMPP → الأقران الخادم
2. **تحرير النظير**:
   - ابحث عن النظير الذي ترغب في تعديله
   - انقر على زر "تحرير"
   - تحديث المعلمات المطلوبة:
     - التغييرات الشائعة: حد TPS، قائمة كلمة المرور، IP البيضاء، المضيف/المنفذ
   - انقر على "حفظ"
3. **تطبق التغييرات على الفور**:
   - يعيد الاتصال الاتصال تلقائيًا بالإعدادات الجديدة
   - لا حاجة لإعادة تشغيل الخدمة
   - انتقل إلى: SMPP → الحالة المباشرة للتحقق
4. **تحقق من التغييرات**:

- تحقق من نجاح الاتصال
- راقب علامة التبويب السجلات للأخطاء
- اختبر تدفق الرسائل إذا كان ذلك مناسبًا

الخيار B: عبر ملف التكوين

المزايا: البنية التحتية ككود، التحكم في الإصدار

الخطوات:

1. **تحرير ملف التكوين**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **تعديل معلمات الربط**:
   - ابحث عن الربط في قائمة `:binds` أو `:server_binds`
   - تحديث المعلمات المطلوبة:
     - التغييرات الشائعة: حدود TPS، قائمة كلمات المرور، IP البيضاء، المضيف/المنفذ
   - مثال:

```
%{
  name: "vodafone_uk",
  # ... معلمات أخرى
  tps_limit: 150,  # تغيرت من 100
  password: "new_password"  # كلمة المرور المحدثة
}
```

3. **حفظ وإعادة تشغيل الخدمة**:

```
sudo systemctl restart omnimessage-smpp
```

4. **تحقق من التغييرات**:
   - انتقل إلى: SMPP → الحالة المباشرة
   - تحقق من نجاح الاتصال

- راقب السجلات للأخطاء
- اختبر تدفق الرسائل

إزالة اتصال

الغرض: إلغاء تشغيل اتصال SMPP

الخطوات:

1. **إخطار المعنيين**:
   - إبلاغ الناقل/الشريك
   - تنسيق نافذة التوقف
2. **فصل الاتصال عبر واجهة الويب**:
   - انتقل إلى: SMPP → الحالة المباشرة
   - ابحث عن الاتصال
   - انقر على "فصل الاتصال"
   - أكد الإجراء
3. **إزالة التكوين**:
   - انتقل إلى: SMPP → الأقران العملاء/الأقران الخادم
   - ابحث عن الاتصال
   - انقر على "حذف"
   - أكد الإزالة
4. **تحقق من الإزالة**:
   - تحقق من الحالة المباشرة - يجب أن يكون الاتصال قد اختفى
   - راجع السجلات لإغلاق نظيف

تمكين وتعطيل الاتصالات

الغرض: أخذ اتصال غير متصل مؤقتًا دون حذف تكوينه

الذي يتحكم في ما إذا كانت نشطة. تحتفظ الأقران المعطلة بجميع `enabled` تمتلك الأقران حقل تكويناتها ولكن لا تقيم أو تقبل الاتصالات.

عبر واجهة الويب:

1. انتقل إلى: SMPP → الأقران العملاء أو الأقران الخادم
2. ابحث عن النظير الذي ترغب في تعطيله
3. انقر على "تحرير"
4. قم بإلغاء تحديد خانة "مفعل"
5. انقر على "حفظ"

سيتم فصل الاتصال على الفور. لإعادة التمكين، كرر الخطوات وتحقق من المربع مرة أخرى.

حالات الاستخدام:

- نوافذ صيانة الناقل المخطط لها
- إيقاف مؤقت لاتصال الشريك أثناء التحقيق
- تعطيل اتصال أثناء انتظار بيانات اعتماد جديدة

سلوك الاتصال

منطق إعادة الاتصال

عندما لا يمكن لنظير العميل الاتصال، أو يتم رفض ربطه، أو يفقد جلسة قائمة لأي سبب، يقوم الجسر بإعادة الاتصال تلقائيًا:

- **التأخير**: تكون المحاولة الأولى بعد 1 ثانية، ثم يتضاعف الانتظار في كل فشل متتالي (2 ث، 4 ث، 8 ث ...) حتى يصل إلى حد أقصى قدره 30 ثانية
- **إعادة تعيين التأخير**: يعود الانتظار إلى 1 ثانية بمجرد أن تبقى الجلسة مرتبطة لمدة 30 ثانية. وبالتالي، يتم إعادة المحاولة بمعدل أبطأ للناقل الذي يقبل الاتصال ثم يسقطه على الفور، وليس كل ثانية
- **متصل يعني مرتبط**: يظهر الربط فقط على أنه "متصل"، ويرسل حركة المرور، نشطًا ولكن لم يصل رد TCP ب��جرد أن يقبل الناقل الربط. بينما يكون اتصال الربط، يظهر الربط على أنه يتصل
- **مهلة الاتصال**: يتم التخلي عن اتصال TCP الذي لا يحصل على إجابة بعد 3 ثوانٍ ويتم إعادة المحاولة مع التأخير المذكور أعلاه
- **العزل**: لا تعيد الاتصالات لنظير واحد بدء تشغيل أقران آخرين أو الخدمة، بغض النظر عن عدد مرات إسقاط الناقل الاتصال

- **بدء التشغيل المتدرج**: عندما تبدأ عدة روابط في وقت واحد (على سبيل المثال، بعد إعادة تشغيل الخدمة)، يتم تدرج الاتصالات مع تأخيرات قدرها 500 مللي ثانية بين كل ربط لتجنب إغراق الشبكة
- **بدء التشغيل المرن**: إذا كان الناقل غير متاح عند بدء تشغيل الجسر، يبدأ الجسر بنجاح ويعيد محاولة الاتصال في الخلفية

تتبع إعادة الاتصال باستخدام مقياس `smpp_reconnect_attempts_total` (انظر MONITORING.md).

فصل نظيف

عندما يتم إيقاف نظير العميل بواسطة "فصل الاتصال" أو عن طريق إيقاف الخدمة أو إعادة تشغيلها، يرسل الجسر `unbind` من نوع SMPP إلى الناقل وينتظر حتى 2 ثانية للحصول على `unbind_resp` قبل إغلاق اتصال TCP. يحرر الناقل الربط على الفور بدلاً من الانتظار حتى تنتهي مهلة keepalive الخاصة به. هذا مهم للناقلات التي تسمح فقط برابط واحد لكل حساب، والتي سترفض إعادة الاتصال بعد إعادة التشغيل.

يتم إغلاق الربط الذي لا يزال يتصل (TCP نشط، الربط لم يتم قبوله بعد) بدون `unbind`.

استعلام الرابط (Keepalive)

يرسل الجسر PDUs من نوع SMPP `enquire_link` بشكل دوري للتحقق من أن الاتصالات نشطة:

- **الفترة الافتراضية**: 60 ثانية (قابلة للتكوين لكل ربط عبر `enquire_link_interval`)
- **تعطيل**: تعيين `enquire_link_interval: 0` (غير موصى به)
- **كشف الفشل**: إذا توقف النظير البعيد عن الاستجابة لـ enquire_link، يعتبر الاتصال ميتًا وتبدأ ��عادة الاتصال

راقب صحة استعلام الرابط عبر مقاييس Prometheus `smpp_enquire_link_sent_total` و `smpp_enquire_link_received_total`. تشير الفجوة المتزايدة بين المرسلة والمستلمة إلى مشاكل في الاتصال.

تحديد معدل TPS

يفرض كل ربط حد `tps_limit` الخاص به باستخدام نافذة متحركة لكل ثانية:

- يتم عد الرسائل داخل كل نافذة 1 ثانية
- عند الوصول إلى الحد، يتوقف عامل قائمة الانتظار حتى الثانية التالية
- يمكن أن تكون هناك 100 رسالة كحد أقصى في الهواء (في انتظار الرد) لكل ربط في أي وقت
- يتم إعادة تعيين النافذة تلقائيًا في بداية كل ثانية جديدة

إذا كنت ترى انخفاضًا في الإنتاجية، تحقق من:

1. أن `tps_limit` تم تعيينه مرتفعًا بما يكفي لحركتك
2. أن `queue_check_frequency` منخفض بما يكفي للحفاظ على إمداد الأنبوب
3. أن الناقل يستجيب للرسائل بسرعة (الاستجابات البطيئة تقلل من الإنتاجية الفعالة)

إدارة تدفق الرسائل

ا��تحقق من قائمة انتظار الرسائل

الغرض: مراقبة الرسائل المعلقة

الخطوات:

1. **الوصول إلى قائمة الانتظار**:
    - انتقل إلى: علامة التبويب قائمة الانتظار
    - عرض قائمة الرسائل المعلقة

2. **تحقق من تفاصيل الرسالة**:
    - انقر على صف الرسالة
    - مراجعة:
        - رقم الوجهة
        - نص الرسالة
        - SMSC المستهدفة (dest_smsc)
        - محاولات التسليم
        - الحالة
3. **البحث عن رسالة محددة**:
    - استخدم فلتر البحث
    - تصفية حسب الوجهة، المحتوى، أو SMSC

استكشاف الأخطاء في الرسائل العالقة

الأعراض: عدم تسليم الرسائل

الخطوات:

1. **تحقق من حالة الاتصال**:
    - انتقل إلى: SMPP → الحالة المباشرة
    - تحقق من أن الاتصال المستهدف متصل
    - إذا كان مفصولًا، انظر إعادة الاتصال
2. **تحقق من تفاصيل الرسالة**:
    - نتقل إلى: علامة التبويب قائمة الانتظار��
    - اب��ث عن الرسالة العالقة
    - تحقق من تطابق حقل `dest_smsc` مع اسم الاتصال
    - تحقق من الطابع الزمني `deliver_after` (جدولة إعادة المحاولة)
3. **تحقق من محاولات التسليم**:
    - محاولات عالية = فشل متكرر
    - تحقق من السجلات للحصول على رسائل الخطأ

- قد تشير إلى تنسيق غير صالح أو رفض من الناقل

4. (إذا لزم الأمر) **التدخل اليدوي**:
    - اتصل بالناقل للتحقق من المشكلة
    - قد تحتاج إلى إلغاء الرسالة وإعادة تقديمها
    - تحقق مع الفريق الخلفي بشأن مشكلات قائمة الانتظار

تمرير إيصالات التسليم (DLR)

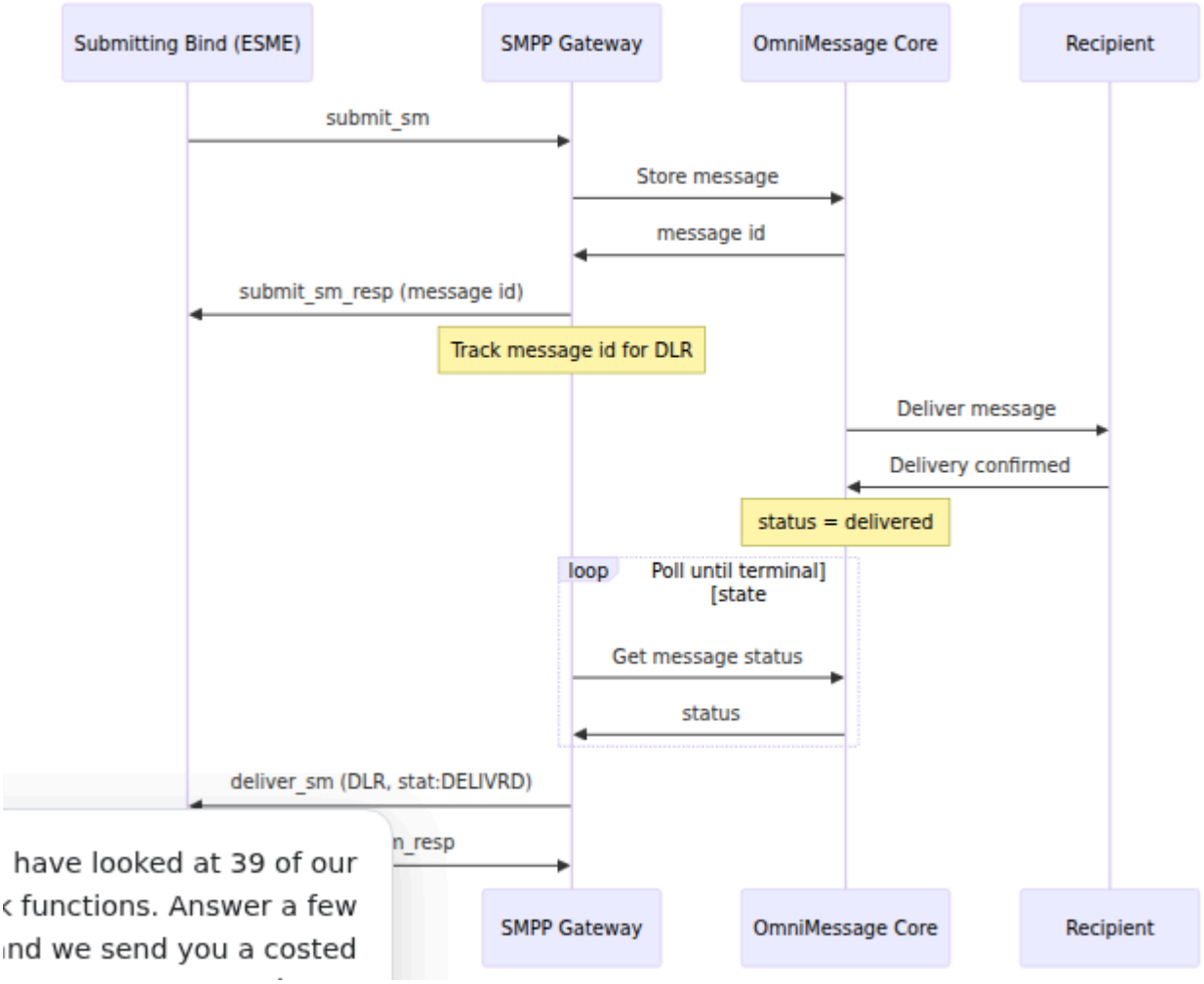
إيصال التسليم (DLR) هو رسالة SMPP يرسل الجسر مقدمة. DLR تبلغ عن الحالة النهائية لرسالة مرة أخرى إلى الربط المقدم عندما يقوم الربط بتعيين `deliver_receipts: true`. استخدم هذه الميزة لربط الاتصال الذي يحتاج مشغلها إلى تأكيد التسليم لكل رسالة.

يرسل الجسر الإيصال كـ PDU من نوع `deliver_sm` مع تعيين `esm_class` `0x04`إ��ى `0` (إيصال تسليم SMSC). يحمل الإيصال نفس معرف الرسالة الذي أعاده الجسر في `submit_sm_resp`، بحيث يمكن للمشغل ربط الإيصال مع الإرسال الأصلي.

المتطلبات

- تعيين `deliver_receipts: true` على الربط الخادم. انظر CONFIGURATION.md.
- تعيين `cache_enabled: false` على نفس الربط. التخزين المؤقت يعيد معرف رسالة مؤقت لا يتطابق مع معرف النواة، لذا فإن الربط المخزن مؤقتًا يكسر ارتباط الإيصالات. يكتب البوابة تحذيرًا عند بدء التشغيل إذا كان الربط يحدد كلاهما.

تدفق الرسالة



قيم حالة الإيصال

حقل الإيصال `stat` يُبلغ عن الحالة النهائية للرسالة.

حالة النواة	إيصال `stat`	المعنى
`delivered`	`DELIVRD`	استلم المستلم الرسالة.
`expired`	`EXPIRED`	انتهت فترة صلاحية الرسالة قبل التسليم.
`dropped`	`UNDELIV`	لم تتمكن البوابة من تسليم الرسالة.
`rejected`	`REJECTD`	رفضت النواة الرسالة.
`balance_rejected`	`REJECTD`	رفضت النواة الرسالة بسبب رصيد الحساب.

ملاحظات تشغيلية

- تتعقب البوابة الرسائل الجارية في الذاكرة. إذا تم إعادة تشغيل الخدمة، فإن الإيصالات للرسائل التي تكون جارية في تلك اللحظة تُفقد. هذه آلية تعتمد على الجهد الأفضل.
- إذا كانت جلسة الربط معطلة عندما تصل رسالة إلى حالة نهائية، تحتفظ البوابة بالإيصال وترسله عند إعادة الاتصال بالجلسة. تسقط البوابة الإيصال بعد 24 ساعة.
- راقب مقاييس DLR. انظر MONITORING.md لتأكيد عمل الميزة.

استكشاف مشاكل الاتصال

إعادة الاتصال بالربط

الأعراض: يظهر الاتصال "Disconnected" (أحمر)

الخطوات:

1. **تحقق من الاتصال بالشبكة**:

```
ping -c 3 carrier-smpp-server.com
telnet carrier-smpp-server.com 2775
```

2. **تحقق من السجلات بحثًا عن الأخطاء**:

- انتقل إلى: علامة التبويب السجلات
- تصفية: مستوى الخطأ
- ابحث عن فشل المصادقة، انتهاء مهلة الشبكة

3. **تحقق من بيانات الاعتماد**:
   - انتقل إلى: SMPP → الخادم / أقران العميل
   - تحقق من أن system_id وكلمة المرور صحيحة
   - اتصل بالناقل إذا كنت غير متأكد
4. **إعادة الاتصال يدويًا**:
   - انتقل إلى: SMPP → الحالة المباشرة
   - ابحث عن الربط المنقطع
   - انقر على زر "Reconnect"
   - انتظر 10-30 ثانية
   - تحقق مما إذا كانت الحالة تتغير إلى "Connected"
   - إعادة الاتصال أيضًا تعيد الربط الذي تم إيقافه مع "Drop"
5. **إذا فشلت إعادة الاتصال**:
   - تحقق من قواعد جدار الحماية
   - تحقق من أن خادم الناقل يعمل
   - اتصل بدعم الناقل
   - انظر TROUBLESHOOTING.md

التعامل مع فشل المصادقة

الأعراض: فشل الربط المتكرر في السجلات

الأسباب:

- اسم المستخدم / كلمة المرور غير صحيحة
- IP غير مدرج في القائمة البيضاء لدى الناقل
- الحساب معلق / منتهي الصلاحية

الخطوات:

1. **تحقق من بيانات الاعتماد**:
    - أقران العميل ← SMPP: انتقل إلى
    - تحقق مرة أخرى من system_id وكلمة المرور
    - تأكد مع الناقل
2. **تحقق من إدراج IP في القائمة البيضاء**:
    - تأكد من IP البوابة لديك مع الناقل
    - اطلب من الناقل التحقق من إدراج IP في القائمة البيضاء
3. **تحقق من حالة الحساب**:
    - تحقق من أن الحساب نشط
    - تحقق من العقود المنتهية
    - اتصل بفوترة الناقل
4. **تحديث التكوين**:
    - إذا تم تغيير بيانات الاعتماد، قم بتحديثها في واجهة الويب
    - انقر على "Reconnect" لإعادة المحاولة مع بيانات الاعتماد الجديدة

المراقبة والتنبيه

التحقق من مقاييس Prometheus

تحقق سريع:

```
curl http://localhost:4000/metrics | grep smpp_connection_status
```

الناتج المتوقع:

```
smpp_connection_status{bind_name="vodafone_uk",...} 1
smpp_connection_status{bind_name="att_us",...} 1
```

يجب أن تكون جميع القيم `1` (متصل).

الاستجابة للتنبيهات

تنبيه انقطاع الاتصال:

1. تحقق من واجهة الويب → SMPP → الحالة المباشرة
2. حاول إعادة الاتصال يدويًا
3. تحقق من السجلات بحثًا عن الأخطاء
4. اتصل بالناقل إذا كان الانقطاع مطولًا
5. انظر TROUBLESHOOTING.md

تنبيه معدل الفشل العالي:

1. تحقق من السجلات بحثًا عن أنماط الأخطاء
2. راجع التغييرات الأخيرة في التكوين
3. اتصل بالناقل حول الرفض
4. تحقق من الامتثال لتنسيق الرسالة

تنبيه عدم وجود حركة مرور:

1. تحقق من أن قائمة الانتظار الخلفية تحتوي على رسائل
2. تحقق من أن توجيه `dest_smsc` صحيح

3. تحقق من أن حدود TPS ليست صارمة جدًا
4. راجع إعداد `queue_check_frequency`

إجراءات الصيانة

الصيانة الروتينية

قم بإجراءها شهريًا:

1. **مراجعة المقاييس**:
    - تحليل اتجاهات حجم الرسائل
    - تحقق من معدلات نجاح التسليم
    - تحديد فرص التحسين
2. **تحديث الوثائق**:
    - وثق أي تغييرات في التكوين
    - تحديث معلومات الاتصال
    - ملاحظة نوافذ صيانة الناقل
3. **تدقيق بيانات الاعتماد**:
    - مراجعة جميع كلمات مرور SMPP
    - التخطيط لدورة بيانات الاعتماد
    - التحقق من أن قوائم IP البيضاء محدثة
4. **تخطيط السعة**:
    - مراجعة معدلات الرسائل القصوى
    - التحقق من حدود TPS
    - التخطيط ��لنمو

إعادة تشغي�ل الخدمة

عند الحاجة:

- بعد تغييرات في ملف التكوين
- بعد تحديثات النظام
- أثناء استكشاف الأخطاء

الخطوات:

```
# تحقق من الحالة الحالية
sudo systemctl status omnimessage-smpp

# إعادة تشغيل الخدمة
sudo systemctl restart omnimessage-smpp

# تحقق من إعادة التشغيل
sudo systemctl status omnimessage-smpp

# تحقق من السجلات
sudo journalctl -u omnimessage-smpp -n 50
```

تحقق عبر واجهة الويب:

1. الوصول إلى لوحة التحكم (قد يستغرق 30-60 ثانية لتصبح متاحة)
2. انتقل إلى: SMPP → الحالة المباشرة
3. انتظر حتى يتم إنشاء جميع الاتصالات (1-2 دقيقة)
4. تحقق من السجلات بحثًا عن الأخطاء

نسخ احتياطي للتكوين

قم بعمل نسخة احتياطية من الملفات الحيوية قبل التغييرات:

```
# نسخ احتياطي للتكوين
sudo cp /opt/omnimessage-smpp/config/runtime.exs \
  /opt/omnimessage-smpp/config/runtime.exs.backup.$(date +%Y%m%d)

# نسخ احتياطي للشهادات
sudo tar -czf /tmp/smpp-certs-$(date +%Y%m%d).tar.gz \
  /opt/omnimessage-smpp/priv/cert/
```

استعادة إذا لزم الأمر:

```
# استعادة التكوين
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup.YYYYMMDD \
  /opt/omnimessage-smpp/config/runtime.exs

# إعادة تشغيل الخدمة
sudo systemctl restart omnimessage-smpp
```

إجراءات الطوارئ

انقطاع الخدمة الكاملة

الخطوات:

1. **تحقق من حالة الخدمة**:

   ```
   sudo systemctl status omnimessage-smpp
   ```

2. **إذا توقفت الخدمة، ابدأها**:

   ```
   sudo systemctl start omnimessage-smpp
   ```

3. **تحقق من السجلات لمعرفة سبب التعطل**:

   ```
   sudo journalctl -u omnimessage-smpp -n 100
   ```

4. **إذا لم تبدأ**:

   - تحقق من أخطاء بناء جملة التكوين
   - تحقق من وجود شهادات SSL
   - تحقق من مساحة القرص: `df -h`
   - تحقق من الذاكرة: `free -h`

5. إذا لم يتم حل المشكلة **اتصل بالدعم**

طلبات الناقل لفصل الطوارئ

الخطوات:

1. **افصل الاتصال على الفور**:
    - انتقل إلى: SMPP → الحالة المباشرة
    - ابحث عن الاتصال المتأثر
    - انقر على "Drop Connection"
2. **وثق السبب**:
    - لاحظ اسم الناقل
    - سجل الوقت والسبب
    - احفظ المراسلات
3. **تحقق من المشكلة**:
    - تحقق من أنماط الرسائل الأخيرة
    - راجع السجلات بحثًا عن الأخطاء
    - حدد السبب الجذري
4. **تنسيق الحل**:
    - العمل مع الناقل
    - تنفيذ الإصلاحات
    - اختبار قبل إعادة الاتصال

ارتفاع حجم الرسائل

الأعراض: حركة مرور رسائل مرتفعة بشكل غير متوقع

الخطوات:

1. **تحقق من حدود TPS**:
    - انتقل إلى: SMPP → الحالة المباشرة
    - تحقق من أن الاتصالات لا تعاني من تقييد
    - قد تحتاج إلى زيادة حدود TPS مؤقتًا

2. **راقب استقرار الناقل**:
    - راقب الانقطاعات
    - تحقق من معدلات نجاح التسليم
3. **تنسيق مع الخلفية**:
    - تحقق من أن مصدر الرسالة شرعي
    - قد تحتاج إلى تنفيذ تحديد المعدل في المنبع
4. **قم بالتوسع إذا لزم الأمر**:
    - قد تحتاج إلى حالات بوابة إضافية
    - اتصل بالدع�� للحصول على نصائح حول التوسع

أفضل الممارسات

قائمة التحقق اليومية

- [ ] تحقق من أن جميع اتصالات SMPP متصلة
- [ ] مراجعة سجلات الأخطاء لأي مشاكل
- [ ] مراقبة قائمة انتظار الرسائل للتراكم
- [ ] تحقق من لوحات معلومات Prometheus/Grafana
- [ ] تأكد من أن معدلات نجاح التسليم > 98%

المهام الأسبوعية

- [ ] مراجعة اتجاهات المقاييس
- [ ] تحقق من الأنماط الشاذة
- [ ] اختبار إجراءات استعادة الكوارث
- [ ] تحديث الوثائق حسب الحاجة
- [ ] مراجعة والاعتراف بالتنبيهات

المهام الشهرية

- [ ] تدقيق بيانات الاعتماد
- [ ] مراجعة تخطيط السعة
- [ ] تحديث جهات الاتصال مع الناقل
- [ ] مراجعة وتحسين إعدادات TPS
- [ ] نسخ احتياطي لملفات التكوين

الوثائق ذات الصلة

- **CONFIGURATION.md** - تكوين الاتصالات والإعدادات
- **SOURCE_ADDRESS_WHITELIST.md** - تقييد العناوين الأصلية لكل نظير خادم
- **MONITORING.md** - إعداد تنبيهات Prometheus
- **TROUBLESHOOTING.md** - حل المشكلات الشائعة
- **README.md** - نظرة عامة على النظام
