

安裝

安裝說明書

安裝

OmniMessage SMPP 安裝說明書 SMPP 安裝 OmniMessage 安裝說明書
OmniMessage Core 安裝 - 安裝說明書

1. 安裝 OmniMessage SMPP PDU
2. 安裝 OmniMessage REST API
3. 安裝 OmniMessage 安裝說明書
4. 安裝 SMPP PDU 安裝說明書
5. 安裝 OmniMessage

OmniMessage 安裝DiameterMAPIMS安裝說明書 - 安裝說明書
OmniMessage Core

安裝

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

安裝說明書

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

安裝

安裝 Elixir 安裝說明書

```
import Config

# 配置
config :omnimessage_smpp,
  setting_name: value

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

配置

API_BASE_URL

OmniMessage Core 配置 URL

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

配置	配置	配置	配置
api_base_url	URL (URL)		-

OmniMessage Core 配置 URL 配置 REST API 配置 OmniMessage 配置

- 配置 SMPP 配置 OmniMessage 配置
- 配置 SMPP 配置
- 配置 OmniMessage
- 配置

配置“配置”配置 OmniMessage 配置

- ✓ 配置 SMPP 连接
- ✓ 配置 OmniMessage 连接
- ✓ 配置 HTTP 连接
- ✓ 配置 SMTP 连接
- ✓ 配置 IMAP 连接
- ✓ 配置 POP3 连接
- ✓ 配置 OAuth2 认证

配置 SMPP ↔ OmniMessage 连接

配置

```
# 本地 IP 上 HTTPS
api_base_url: "https://192.168.1.100:8443"

# 远程 HTTPS
api_base_url: "https://omnimessage-core.company.com:8443"

# HTTP 连接
api_base_url: "http://192.168.1.100:8080"
```

配置

- 配置 OmniMessage Core
- 配置 HTTPS 连接 `verify_ssl_peer`
- 配置 OAuth2 认证

SMPP_POLL_INTERVAL

配置

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


SMSC_NAME

配置示例

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

属性	类型	默认值	说明
smpp_name	String		"smpp_gateway"

配置说明

配置项 SMSC_NAME

SMPP 配置

配置项 ESME 和 SMSC 用于配置 SMPP 连接。ESME 是发送端，SMSC 是接收端。配置项 ESME 和 SMSC 用于配置 SMPP 连接。ESME 是发送端，SMSC 是接收端。

配置 SMPP 连接

```
config :omnimessage_smpp, :binds, [
  %{
    # 连接名称
    name: "vodafone_uk",

    # 模式
    mode: :client,

    # SMPP 连接类型
    bind_type: :transceiver,

    # SMPP 主机地址
    host: "smpp.vodafone.co.uk",
    port: 2775,

    # 系统 ID
    system_id: "your_username",
    password: "your_password",

    # SMPP 系统类型
    system_type: "",
    addr_ton: 0,
    addr_npi: 0,
    address_range: "",

    # TPS 限制
    tps_limit: 100,

    # 队列检查频率
    queue_check_frequency: 1000,

    # 链路询问间隔
    enquire_link_interval: 60,

    # 缓存配置
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60
  }
]
```

CarrierRegion

name

CarrierRegion

CarrierRegion	CarrierRegion	CarrierRegion
CarrierRegion	CarrierRegion	"vodafone_uk"

CarrierRegion SMPP CarrierRegion

- CarrierRegion
- CarrierRegion
- CarrierRegion

CarrierRegion

- carrier_region "vodafone_uk" "att_us"
- purpose_number "marketing_1" "alerts_primary"

mode

CarrierRegion

CarrierRegion	CarrierRegion	CarrierRegion
CarrierRegion	CarrierRegion	:client

CarrierRegion ESME CarrierRegion SMSC

CarrierRegion :client

bind_type

SMPP CarrierRegion

項目	値	説明
項目	値	:transmitter, :receiver, :transceiver

項目項目項目項目項目項目

項目

- :transmitter - 項目項目submit_sm
- :receiver - 項目項目deliver_sm
- :transceiver - 項目項目項目項目

項目項目項目項目項目項目項目項目項目項目 :transceiver

host

項目 **SMPP** 項目項目項目 **IP**

項目	項目	項目
項目	項目	"smpp.carrier.com" 項目 "10.5.1.100"

項目項目項目 **SMPP** 項目項目項目

項目

```

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

```

port

SMPP 項目項目

項目	項目	項目	項目
項目	項目	2775	1-65535

SMPP TCP

2775

port: 2775 #
port: 3000 #

system_id

		"company_user"

-

password

		"secret_password"

-
-
-

tps_limit

配置参数

参数	单位	默认值	范围
tps_limit	TPS	100	1-10000

配置参数说明

说明

- tps_limit 70-80%
- 单位/秒
- 配置参数

配置

```
tps_limit: 10    # 配置  
tps_limit: 50    # 配置  
tps_limit: 100   # 配置  
tps_limit: 1000  # 配置
```

配置

```
tps_limit = 100 TPS  
tps_limit = 70-80  
20-30 TPS 配置
```

queue_check_frequency

配置参数

参数	单位	默认值	范围
queue_check_frequency	Hz	1000	100-10000

配置参数说明

TPS

- TPS>**100 TPS**500-1000ms
- TPS**10-100 TPS**1000-2000ms
- TPS<**10 TPS**2000-5000ms

API

- API = API
- API = API

enquire_link_interval

SMPP

		60	0-3600

SMPP enquire_link PDU enquire_link_resp

enquire_link

- 60
- 15-30
- 120-300
- 0 enquire_link

enquire_link

```
enquire_link_interval: 60 # 1
enquire_link_interval: 30 # 
enquire_link_interval: 0  # 
```

system_type

SMPP

- 1 - ISDN/E.164
- 3 - 3GPP/X.121
- 9 - 其他

地址范围

address_range

配置项

名称	类型	默认值	说明
address_range	字符串	"614*"	

该配置项用于指定 SMPP 地址范围，默认为 "614*"。

enabled

配置项

名称	类型	默认值
enabled	布尔值	true

该配置项用于指定是否启用 SMPP 功能，默认为 true。

cache_enabled

配置项

名称	类型	默认值
cache_enabled	布尔值	false

该配置项用于指定是否启用 API 消息缓存功能，默认为 false。
[MESSAGE_CACHE.md](#)

cache_max_size

配置項目

項目	単位	初期値	範囲
キャッシュサイズ	MB	10000	1-1000000

キャッシュの有効化/無効化を制御するパラメータ。FIFO方式で、cache_enabled が true の場合

cache_retry_interval

設定項目

項目	単位	初期値
キャッシュリトライ間隔	秒	60

キャッシュリトライの待ち時間。0は60秒、1は120秒、2は240秒。cache_enabled が true の場合

Web UI 設定

SMPP 配置指南

本文档介绍了如何配置 SMSC 和 ESME 的 SMPP 连接。配置文件中包含以下部分：

配置示例：

```
config :omnimessage_smpp, :server_binds, [
  %{
    # 合作伙伴名称
    name: "partner_acme",

    # 系统 ID
    system_id: "acme_corp",
    password: "acme_secret",

    # 允许的绑定类型
    allowed_bind_types: [:transmitter, :receiver, :transceiver],

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

    # 源地址白名单 = 空列表
    source_address_whitelist: [],

    # TPS 限制
    tps_limit: 50,

    # 队列检查频率
    queue_check_frequency: 1000,

    # 链路轮询间隔
    enquire_link_interval: 60,

    # 缓存配置
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60
  }
]
```

XXXXXXXXXX

name

XXXXXXXXXX

id	name	value
0000	1	"partner_acme"

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXX/XXXXXXXXXXXX

system_id

XXXXXXXXXX

id	name	value
0000	1	"acme_corp"

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

password

XXXXXXXXXX

id	name	value
0000	1	"secure_password"

XXXXXXXXXXXXXXXXXXXX

XXXX

- 接收器
- 发射器
- 收发器

allowed_bind_types

绑定类型

绑定类型	默认值	说明
接收器	否	-

接收器、发射器、收发器

示例

```
allowed_bind_types: [:transceiver] # 收发器
allowed_bind_types: [:transmitter, :receiver] # TX 和 RX
allowed_bind_types: [:transmitter, :receiver, :transceiver] # 全部
```

接收器、发射器、收发器

ip_whitelist

IP 地址列表

IP 地址	默认值	说明
IP 地址	否	IP 地址 CIDR 格式

IP 地址 - IP 地址

示例

- IP "192.168.1.100" /32
- CIDR "192.168.1.0/24" 或 "10.0.0.0/8"
- ["192.168.1.0/24", "10.50.1.100"]

例

```
# 空 IP 白名单
ip_whitelist: []

# 1 IP
ip_whitelist: ["203.0.113.50"]

# 2 IP
ip_whitelist: ["203.0.113.50", "203.0.113.51"]

# CIDR
ip_whitelist: ["192.168.1.0/24"]

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

注

- `/32` - 1 IP 白名单 IP 1 台
- `/24` - 256 台 IP 白名单 192.168.1.0-255
- `/16` - 65,536 台 IP 白名单 10.50.0.0-255.255
- `/8` - 16,777,216 台 IP 白名单 10.0.0.0-255.255.255.255

source_address_whitelist

例

名前	型	例	説明
source_address_whitelist	list	[""]	source IP 白名单

source_address_whitelist には、source IP の ID を指定します。

例

- source IP `"MyBrand"` を白名单 `"MyBrand"`
- source IP `"614*"` を白名单 `"614"` の子として

cache_enabled

□□□□□□□□

□□□□□□□ □□□□□□□□□□□□□□□ □□□□□□□□□□ MESSAGE_CACHE.md□

cache_max_size

□□□□□□□□

□□□□□□□□ □□□□□□□□□□ □□□

cache_retry_interval

□□□□□□□□□□

□□□□□□□□ □□□□□□□□□□□□□□ □□□

Web UI □□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

配置

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

主机

host

主机 IP 地址

名称	IP 地址	主机名	备注
主机	0.0.0.0	0.0.0.0 127.0.0.1	

主机 IP 地址

主机

- 0.0.0.0 - 主机 IP 地址
- 127.0.0.1 - 本地主机 IP 地址
- 192.168.1.10 - 主机 IP 地址

port

主机 TCP 端口

名称	端口	端口范围	备注
主机	2775	1-65535	

主机 SMPP 端口

主机 2775

max_connections

配置参数

参数名	数据类型	默认值	取值范围
max_connections	int	100	1-10000

配置参数说明

说明

- 配置参数名称
- 配置参数单位
- 配置参数范围

□□□□□□

□□ **1**□□□□□□□□

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.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
  }
]
```

2

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.companv.com:8443"

config :omnimessage_smpp, :binds, [
  #
  %{
    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
  },

  #
  %{
    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
  }
]
```

03 3 0000000000000000

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.companay.com:8443"

# 0000
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
  }
]

# 00000000
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
  }
]
```

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

配置

配置

配置

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

配置

配置

```
# 配置
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-smpp console
```

按 **Ctrl+C** 退出

配置

配置

環境変数	設定項目	値
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
MNESIA_STORAGE_TYPE	mnesia_storage_type	disc_copies

systemd の設定

```
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"
```

インストール

1. パッケージのインストール

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

2. 設定ファイルの編集

- 12 行目
- 13 行目
- 14 行目

3. IP の設定

- 15 行目 `ip_whitelist`
- 16 行目 `[]`

4. SSL の設定

- 17 行目 `verify_ssl_peer: true`

5. ログの設定

- 18 行目
- 19 行目

README

- [MONITORING.md](#) 監視
- [USAGE.md](#) 使用方法
- [TROUBLESHOOTING.md](#) トラブルシューティング
- [README.md](#) 本 README



□□□□

A

API□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□

Web UI □□□□□□□□□□□□□□□□□□

B

□□

SMPP □□□□□□□□□□□□□□□□

□□

□□□□□□ SMPP □□□□□□□□□□□□□□

□□□□

SMPP □□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□

SMPP □□□□□□□□□□□□□□ IP □□□

C

CIDR□□□□□□□□

□□□□ IP □□□□□□□□□□□□ 192.168.1.0/24 □□ 256 □ IP □□□□

SMPP ESME SMSC SMPP

SMPP □□□□□□□□

- 
- 
- 

□ □

D

□□□□□□□□ SMPP □□□GSM-7□UCS-2 □□□

SMSC **SMPP PDU** **ESME**

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □

□□□□□□□□□□ SMPP □□□□□□□□□□

[illegible]

E

0000

00000 SMPP 00000000000000000000

ESM 0

0000000000 SMPP 000

ESME0000000000

0 SMPP 0000000 SMSC 000000000000000000000000 00000 00000000 ESME 0000000
SMSC00000 00000 0000000000000 ESME 00000

0000

000000000000000000000000100020004000800.....00

F

000

00000000000000000000

0000

SMPP 000 OmniMessage Core 00000000 60 0000000000000000000000000000000000 90 000
00OmniMessage 000000000000

G

00

0000000000000000000000 SMPP 00000000

00

0000000000000000000000000000000000

Grafana

0000000000 Prometheus 000000000000

GSM-7

00 SMS 000 7 00000000000000000000 160 0000

H

HTTP/HTTPS

Web 透過 HTTPS 通訊

I

IP 位址

透過 IP 位址進行通訊

ISDN 數位式網路

數位式網路通訊

J

()

K

enquire_link

KPI 效能指標

效能指標

L

Prometheus bind_name="vodafone_uk"

Web UI Phoenix

M

消息

消息队列消息队列消息队列

消息

消息队列消息队列 Prometheus 消息队列

MO消息队列

消息队列消息队列消息队列

MT消息队列

消息队列消息队列消息队列

MSISDN消息队列消息队列消息队列

消息队列消息队列

N

NPI消息队列消息队列

消息队列消息队列 SMPP 消息队列消息队列ISDN消息队列

O

消息

消息队列消息队列消息队列

消息

消息队列消息队列消息队列

P

PDU消息队列消息队列

消息 SMPP 消息队列消息队列submit_sm消息队列deliver_sm消息队列

Prometheus

配置 Prometheus 监控

Q

问题

配置 Prometheus 监控

配置 Prometheus

配置 Prometheus 监控

配置 Prometheus

配置 Prometheus 监控 SMPP 接口

R

配置 Prometheus

配置 Prometheus 监控 TPS

配置 Prometheus

配置 Prometheus 监控 SMPP 接口 deliver_sm

配置 Prometheus

配置 Prometheus 监控 SMPP 接口

配置 Prometheus

配置 Prometheus 监控

S

配置 Prometheus

配置 Prometheus 监控 SMPP PDU 接口 submit_sm 和 submit_sm_resp

配置 Prometheus

配置 Prometheus 监控 ESME 和 SMSC 接口

□□□

□□□□□□ SMPP □□□□□submit_sm□□

□□□

□□□□□□□□□□□□□□□□□□

U

UCS-2

□□ SMS □ 16 □ Unicode □□□□□□□□□□□□ 70 □□□□

□□□□□□

□□□□□□□□□□□□□□□□

V

□□□

□□□□□□□□□□□□□□□□□□□□

W

Web □□□

□□□□□□□□□□□□□□□□□□□□

□□□

□ IP □□□□□□□□□□□□

X

(□□□□)

Y

(□□□□)

Z

()

□□□□□□□□

□□□	□□
API	□□□□□□□□
CIDR	□□□□□□
DLR	□□□□
ESME	□□□□□□□□
GSM	□□□□□□□□
HTTP	□□□□□□□□
HTTPS	□□□□□□□□□□
IP	□□□□□□
ISDN	□□□□□□□□
KPI	□□□□□□□□
MO	□□□□□
MSISDN	□□□□□□□□□□□□
MT	□□□□□
NPI	□□□□□□□□
PDU	□□□□□□□□
SMPP	□□□□□□□□
SMSC	□□□□□□□□

項目	説明
SMS	短メッセージサービス
SSL	セキュアソケットレイヤー
TLS	トランスポートレイヤーセキュリティ
TON	トーン
TPS	トランザクション処理システム
UCS	ユニバーサル通信システム
UI	ユーザーインターフェース
URL	ユニークリソースロケータ

目次

- **README.md** - 概要
 - **CONFIGURATION.md** - 設定
 - **USAGE.md** - 使い方
 - **MONITORING.md** - 監視
 - **TROUBLESHOOTING.md** - トラブルシューティング
-

SMPP 架构图

简介

SMPP 架构图展示了 SMPP 客户端 API 与 Mnesia 数据库的交互流程。API 通过 Mnesia 数据库与 Backend API 进行交互。

组件

- SMPP 客户端 - 通过 API 与 SMPP 服务器交互
- API - 通过 Mnesia 数据库的 `:disc_copies` 与 Mnesia 数据库交互
- Mnesia 数据库 - 存储 SMPP 客户端的元数据
- Backend API - 通过 SMPP 客户端与 SMPP 服务器交互
- Cache - 存储 SMPP 客户端的元数据，采用 FIFO 策略
- CacheFlushWorker - 负责将缓存数据写入 Mnesia 数据库
- LiveView 组件 - 用于实时监控 SMPP 客户端的状态
- Prometheus** 组件 - 用于监控 SMPP 客户端的性能

数据流

提交消息

接收消息

SMPP Client → submit_sm → Gateway Server

↓

Cache in Mnesia (immediate response)

↓

CacheFlushWorker (background)

↓

Backend API

□□□□□

SMPP Client → submit_sm → Gateway Server

↓

Backend API (direct)

□□

1. **MessageCache** □□ (lib/sms_c/smpp/message_cache.ex)

- □□□□□□
- □□□□
- □□ LiveView □□□□□□□□□□

2. **CacheFlushWorker** (lib/sms_c/smpp/cache_flush_worker.ex)

- □□□□□□□□□□ GenServer
- □□□□□□□□□□
- □□□□□□
- □□□□□□□□□□

3. **Mnesia** □ (:smpp_message_cache)

- □□ :disc_copies □□□□□
- □ bind_name□next_retry_at □ status □□
- □□□□□□□□□□□□

□□

□□□□□

□□ config/runtime.exs□

```

config :omnimessage_smpp,
  # 配置缓存
  cache_flush_interval: 10_000,

  # 配置 failed_permanent 重试次数
  cache_max_retry_attempts: 10,

  # 配置缓存
  cache_backoff_multiplier: 2,

  # Mnesia 存储 (:disc_copies 或 :ram_copies)
  mnesia_storage_type: :disc_copies

```

配置 SMPP 连接

配置 SMPP 连接

```

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

    # 配置
    cache_enabled: true,           # 是否启用缓存
    cache_max_size: 10_000,       # 缓存最大大小
    cache_retry_interval: 60      # 缓存重试间隔
  }
]

```

配置

```

# 配置
CACHE_FLUSH_INTERVAL=10000          # 配置缓存
CACHE_MAX_RETRY_ATTEMPTS=10          # 配置 failed_permanent 重试次数
CACHE_BACKOFF_MULTIPLIER=2           # 配置缓存
MNESIA_STORAGE_TYPE=disc_copies      # Mnesia 存储

```

メッセージ

キュー

メッセージキューの仕組み

```
メッセージ60s  
メッセージ2
```

```
キュー 0: 60s (1メッセージ)  
キュー 1: 120s (2メッセージ)  
キュー 2: 240s (4メッセージ)  
キュー 3: 480s (8メッセージ)  
キュー 4: 960s (16メッセージ)  
キュー 5: 1920s (32メッセージ)  
...  
キュー 9: 30,720s (8.5日)
```

失敗

キュー 10 のメッセージが失敗した `failed_permanent` の状態

- メッセージが失敗した
- キューが壊れた
- メッセージが“永続的に失敗”した
- メッセージが削除された

メッセージ

```
:pending → :delivering → SUCCESS (メッセージ)  
                → FAILURE → :pending (メッセージ)  
                        → :failed_permanent (メッセージ)
```



LiveView 简介

通过 `http://your-server:4000/smpp` → “接收”数据

接收数据

- 接收 - 接收数据
- 接收 - 接收数据
- 接收 - 接收数据

接收数据

- 接收
- 接收
- 接收 / 接收
- 接收
- 接收

Prometheus 简介

```
# 接收数据
smpp_cache_size{bind_name="my_bind",mode="server"} 42

# 接收数据
smpp_cache_delivered_total{bind_name="my_bind"} 1234

# 接收
smpp_cache_retry_total{bind_name="my_bind"} 56

# 接收
smpp_cache_permanent_failures_total{bind_name="my_bind"} 2

# 接收
smpp_cache_overflow_total{bind_name="my_bind"} 0
```

□□□□

```
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
```

□□

□□/□□□□

□□ **LiveView UI**

1. □□□ `http://your-server:4000/smpp`
2. □□“□□□□□”□“□□□□□”□□□
3. □□□□
4. □□“□□□□□”□□□
5. □□□□

□□ **IEx** □□□

```
# 开启缓存
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
    "password",
    cache_enabled: true,
    cache_max_size: 10_000,
    cache_retry_interval: 60
)

# 关闭缓存
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
    "password",
    cache_enabled: false
)
```

查看缓存

```
# 查看缓存统计
SmsC.SMPP.MessageCache.get_cache_summary()
# => %{total_cached: 42, pending: 40, failed: 2}

# 查看指定绑定的缓存
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}
# ]

# 查看指定绑定的缓存消息数量
SmsC.SMPP.MessageCache.count_cached_messages("my_bind")
# => 42
```

清理缓存

清理所有缓存

```
# 删除失败消息
{:atomic, failed_messages} = :mnesia.transaction(fn ->
  :mnesia.match_object({:smpp_message_cache, :_, "my_bind", :_,
    :_, :_, :_, :_, :_, :failed_permanent})
end)

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

删除失败消息

```
# 将 failed_permanent 消息改为 pending
SmsC.SMPP.MessageCache.update_cache_record("my_bind", -123456, %{
  status: :pending,
  retry_count: 0,
  next_retry_at: DateTime.utc_now(),
  last_error: nil
})
```

缓存

缓存 / 消息

缓存 `cache_overflow_total` 消息溢出次数

缓存 消息

缓存

1. 缓存 `cache_max_size`
2. 通过 API 设置缓存大小 API 设置
3. 缓存消息
4. 缓存消息

□□□□□□

□□□ □□□□□ :pending □□

□□□□□

1. API □□□

- □□ API □□□
- □□□□ API □□
- □□□□□□

2. next_retry_at □□□

- □□□□ next_retry_at □□□□□
- □□□□□□□□

3. □□□□□□□□□□

```
# □□□□□□□□□□□□  
Supervisor.which_children(SmsC.SMPP.Supervisor)
```

4. □□□□□

- □□□□□□□□ cache_enabled: true

□□□□□□

□□□ □□□□□□□□ retry_count □

□□□

```
# 原子化操作
{: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)

# 记录最后错误
Enum.each(high_retry, fn {_ , _ , _ , msg_id, _ , _ , retry_count, _ ,
  last_error, _} ->
  IO.puts("Message #{msg_id}: #{retry_count} retries, error: #
    {inspect(last_error)}")
end)
```

Mnesia 安装

安装 Mnesia

安装 Mnesia 依赖

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

安装

1. 安装 Mnesia 依赖
2. 设置 `cache_max_size`
3. 设置 Mnesia 的 FIFO 策略

メモリ

メモリ

- メモリ容量 500-1000 MB程度
- 10,000 項目 $\approx$ 5-10 MB
- `:disc_copies` 設定

CPU

- CPU 10 程度
- メモリ 100 MB程度
- CPU 10 程度 API

I/O

- `:disc_copies` 設定
- `>1000 msg/sec` 程度
 - `:ram_copies` 設定
 - 設定
 - 設定

メモリ

項目	cache_max_size	cache_flush_interval
項目 (<100 msg/sec)	10,000	10,000ms
項目 (100-500 msg/sec)	50,000	5,000ms
項目 (>500 msg/sec)	100,000	3,000ms

--

--	--	--	--	--	--

1. Mnesia `disc_copies`
- 2.
- 3.

□ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □

1. `cache_enabled: false`
2. `cache_max_size: 10,000`
3. `cache_retry_interval: 60`
4. `cache_ttl: 3600`

API □□□□

1. 
2. 
3. 
4. 

1111

1. `cache_size` - `cache_size`
2. `cache_size` - `cache_size` `cache_permanent_failures_total` `cache_overflow_total`
3. `cache_size` - `cache_size` `cache_max_size`
4. `cache_size` - `cache_size`
5. `cache_size` - `API` `cache_size`
6. `cache_size` - `API` `cache_size`
7. `cache_size` - `cache_size` `mnesia_storage_type: disc_copies`



- 
- 
- 

Getting Started

SMPP Configuration

Prerequisites

SMPP requires Prometheus to be installed and running on your system.

OmniMessage Core provides an **OmniMessage** endpoint for Prometheus to scrape metrics.

1. **SMPP** - Install and configure the SMPP client.
2. **OmniMessage API** - Install and configure the OmniMessage API.

Installation

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

Install Prometheus on your system.

Run the following command to start Prometheus:

Usage

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

Configuration

Configure the `smpp_` namespace in your Prometheus configuration file.

□ □ □ □ □

omnmessage_smpp_license_status

□□□Gauge

□□□□□□□□

11

- 1 = □□□□
- 0 = □□/□□□□

□□□□

111

```
omnimessage_smpp_license_status 1
```

111

- 000000000000000000000000
- 00000000000000000000SMPP000000
- Web UI0000000000000000

omnmessage_smpp

111

- `license_status == 0`
- SMPP
-
- UI

□□□□□

```
- alert: SMPP_License_Invalid
  expr: omnimessage_smpp_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "SMPP许可证过期"
    description: "许可证过期 - 请及时续费"
```

配置项

smpp_connection_status

配置项Gauge
配置SMPP许可证过期时间
单位

- 1 = 过期
- 0 = 未过期

配置项

- bind_name - 绑定名称“vodafone_uk”
- mode - 模式“client”或“server”
- host - 主机地址
- port - 端口
- bind_type - SMPP绑定类型
- system_id - 系统ID

配置项

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

配置项

- 0000000000000000
- 0000000000000000
- 00000000

000000

smpp_messages_sent_total

Counter

SMPP0000000000

0000

connection_status

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

- 00000000/00
- 0000/000
- 0000000000

smpp_messages_received_total

Counter

SMPP0000000000

0000

connection_status

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

📄

- 📄📄📄📄
- 📄📄📄📄MO📄📄
- 📄📄📄📄📄📄📄

📄📄📄

smpp_delivery_failures_total

📄📄Counter

📄📄📄📄📄📄📄

📄📄📄

📄📄📄connection_status📄

📄📄

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

📄📄

- 📄📄📄📄
- 📄📄📄📄📄
- 📄📄📄

📄📄📄📄

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

计数器

smpp_bind_success_total

Counter

计数器

计数器

值

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

值

- 计数器
- 计数器

smpp_bind_failures_total

Counter

计数器

计数器

值

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

值

- 计数器
- 计数器
- 计数器

计数器

smpp_connection_attempts_total

Counter

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

-
-

smpp_disconnection_total

Counter

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

-
-
-

smpp_enquire_link_sent_total

Counter

PDU

PDU

connection_status

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

-
-

smpp_enquire_link_received_total

Counter

PDU

PDU

connection_status

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

- >>
-

smpp_uptime_seconds

Gauge

SMPP

概要

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

概要

- 概要
- 概要
- 概要

概要

概要MESSAGE_CACHE.md

smpp_cache_size

Gauge

概要

概要

概要

- `bind_name` - 概要
- `mode` - 概要“client”“server”

概要

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

概要

- 概要
- `cache_max_size` 概要

smpp_cache_delivered_total

Counter

API

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

smpp_cache_retry_total

Counter

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

smpp_cache_permanent_failures_total

Counter

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

- $x > 0$   

smpp_cache_overflow_total

Counter

000

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

000

- 0000000000000000API00000000

OmniMessage API0000

0000000000SMPP00000000OmniMessage API00000000000000

00OmniMessage0000000000

- omnimessage_api_requests_total - 00000000API0000
- omnimessage_api_request_duration_seconds - API0000
- omnimessage_queue_depth - OmniMessage0000000000

0000000000000000

0000000000API0000

- "api.*connection refused" - 000000OmniMessage
- "api.*timeout" - OmniMessage000
- "api.*http 503" - OmniMessage00000
- "api.*parse error" - 000000

Prometheus00

00000000

000/etc/prometheus/prometheus.yml0

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

□□□□

```
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'
```

□□□□

□□□□□□□□□□

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

□□/etc/prometheus/targets/smpp-production.json□

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

Grafana

Dashboard

Query

Panel

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

Stat

Options

- `smpp_connection_status < 1`
- `smpp_connection_status == 1`

Table

Panel

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

Graph

Y-axis

Y-axis label: `{{bind_name}}`

00000000

000

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

0000Gauge

000000000-1000

000

- 000 < 95%
- 000 95-98%
- 000 > 98%

000000000000

000

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

0000Stat

000000

00000

Prometheus0000

000 /etc/prometheus/rules/smpp-alerts.yml0

```

groups:
- name: smpp_gateway
  interval: 30s
  rules:
    # 连接
    - alert: SMPPConnectionDown
      expr: smpp_connection_status == 0
      for: 2m
      labels:
        severity: critical
      annotations:
        summary: "SMPP连接{{ $labels.bind_name }}断开"
        description: "连接{{ $labels.bind_name }}已断开2分钟"

    # 失败率
    - alert: SMPPHighFailureRate
      expr: |
        (
          rate(smpp_delivery_failures_total[5m])
          /
          rate(smpp_messages_sent_total[5m])
        ) > 0.05
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: "连接{{ $labels.bind_name }}失败率过高"
        description: "连接{{ $labels.bind_name }}失败率{{ $value |
humanizePercentage }}%"

    # 绑定失败
    - alert: SMPPBindFailures
      expr: increase(smpp_bind_failures_total[10m]) > 3
      labels:
        severity: warning
      annotations:
        summary: "连接{{ $labels.bind_name }}绑定失败"
        description: "连接{{ $labels.bind_name }}在10分钟内发生{{
$value }}次绑定失败"

    # 无流量
    - alert: SMPPNoTraffic
      expr: rate(smpp_messages_sent_total[10m]) == 0

```

```

    for: 30m
    labels:
      severity: warning
    annotations:
      summary: "⚠️ {{ $labels.bind_name }} 连接异常"
      description: "{{ $labels.bind_name }} 30分钟内连接异常"

# 连接异常
- alert: SMPPFrequentDisconnections
  expr: increase(smpp_disconnection_total[1h]) > 5
  labels:
    severity: warning
  annotations:
    summary: "⚠️ {{ $labels.bind_name }} 连接异常"
    description: "{{ $labels.bind_name }} 连接异常 {{ $value }}"

# OmniMessage API 连接异常
- alert: OmniMessageAPIUnreachable
  expr: |
    count(count_over_time({job="omnmessage-smpp"} |=
"api.*connection refused"[5m])) > 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "OmniMessage API 连接异常"
    description: "SMPP 连接异常 OmniMessage API 连接异常 API_BASE_URL 连接异常"

# OmniMessage API 超时
- alert: OmniMessageAPITimeout
  expr: |
    count(count_over_time({job="omnmessage-smpp"} |=
"api.*timeout"[5m])) > 5
  for: 2m
  labels:
    severity: warning
  annotations:
    summary: "OmniMessage API 超时"
    description: "连接异常 API 连接异常 OmniMessage 连接异常"

# 连接异常 API 连接异常
- 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: "送信受信エラー - OmniMessage"
      description: "30分間送信受信エラー発生 OmniMessage API"
```

prometheus.yml

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

Web

Web UI Prometheus



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

SMPP →

-
-
-
-
- 5

- 
- 
- 

□□□□□□

- 000000000000000000000000
- 000000000000000000000000/000000
- 0000000000000000000000000000/000000
- 00

--	--	--	--

□ □ □ □

5/5

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

# 100行
sudo journalctl -u omnimessage-smpp -n 100

# 1時間前以降
sudo journalctl -u omnimessage-smpp --since "1 hour ago"

# エラー
sudo journalctl -u omnimessage-smpp -p err
```

Web UI

Web UI

-
-
-
- /
-

- 時間あたりに送信できるメッセージ数
- 送信成功率
- 送信メッセージ/受信メッセージ
- 遅延/時間あたりに送信できるメッセージ数
- 送信失敗率

送信遅延率KPI

送信遅延率

送信遅延率の計算式

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

送信遅延率 > 99.9%

送信成功率

送信成功率の計算式

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

送信成功率

送信成功率

送信成功率の計算式

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

送信成功率 > 98%

📊 📈 📉 📊

📊 📈 📉 📊

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

📊 < 10 📊 📈 📉 📊

📊 📈 📉 📊

1. 📊 📈

- 📊 📈 Prometheus 📊
- 📊 PagerDuty/OpsGenie 📊 24/7 📊
- 📊 📈

2. 📊 📈

- 📊 📈 Grafana 📊
- 📊 📈 📊 📈
- 📊 📈

3. 📊 📈

- 📊 📈
- 📊 📈
- 📊 📈

4. 📊 📈

- 📊 📈
- 📊 📈 TPS 📊
- 📊 📈 / 📊

5. 配置

- 配置API
 - 配置参数
 - 配置
-

配置参数

配置

配置 `smpp_connection_status`

- 0 = 配置
- 配置 = 配置

配置

配置 `smpp_delivery_failures_total`

- 配置 = 配置
- 配置 = 配置

配置

配置 `smpp_messages_sent_total`

- 配置 = 配置TPS
- 配置API

配置

配置 `smpp_bind_failures_total`

- 配置 = 配置
- 配置 `system_id`

□□□□

- **CONFIGURATION.md** - □□□□□□
 - **USAGE.md** - □□□□
 - **TROUBLESHOOTING.md** - □□□□
 - **README.md** - □□□□□□□□
-

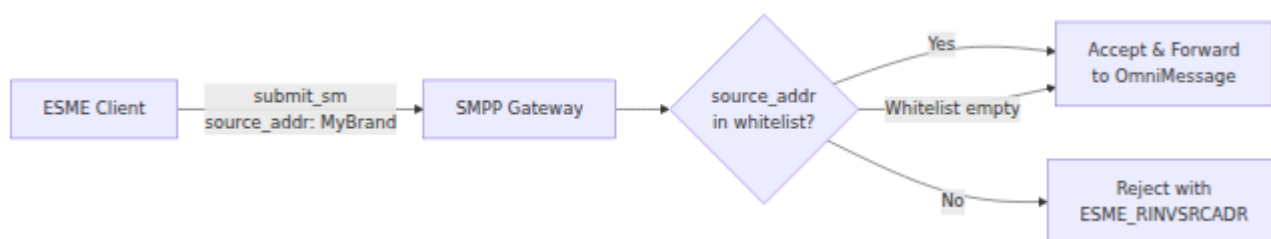
ESME

ESME SMPP source_addr

ESME

ESME SMPP submit_sm PDU PDU CLI / ID source_addr

IP



ESME

ESME

ESME

ESME

電話番号	名前	性別
MyBrand	MyBrand	男
MyBrand	mybrand	男
MyBrand	MyBrands	男
+61400000001	+61400000001	男

電話番号を指定する

電話番号を指定する * 電話番号を指定する

電話番号	名前	性別
614*	61400000001	男
614*	61412345678	男
614*	61500000001	男
+614*	+61400000001	男
My*	MyBrand	男
My*	MyCompany	男

名前

電話番号を指定する * 電話番号を指定する

電話番号を指定する MyBrand, 614*, +61400000001

Source Address	Destination Address	Port
MyBrand	MyBrand 10.10.10.10	22
61412345678	614* 10.10.10.10	22
+614000000001	+614000000001 10.10.10.10	22
OtherBrand		22
615000000001		22

10.10.10.10

10.10.10.10 submit_sm 10.10.10.10

Field	Value
PDU	submit_sm_resp
Length	0x0000000A
ESME_RINVSRCADR	10.10.10.10
ESME_RINVSRCID	

10.10.10.10 10.10.10.10 10.10.10.10

SMPP Server: Rejected submit_sm from partner_acme - source_addr 'UnauthorisedBrand' not in whitelist



Web UI

1. **SMPP > Server Peers**
2.
3. IP
4.

MyBrand,614*,+61400000001

5.

submit_sm PDU

runtime.exs source_address_whitelist

```
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
  }
]
```

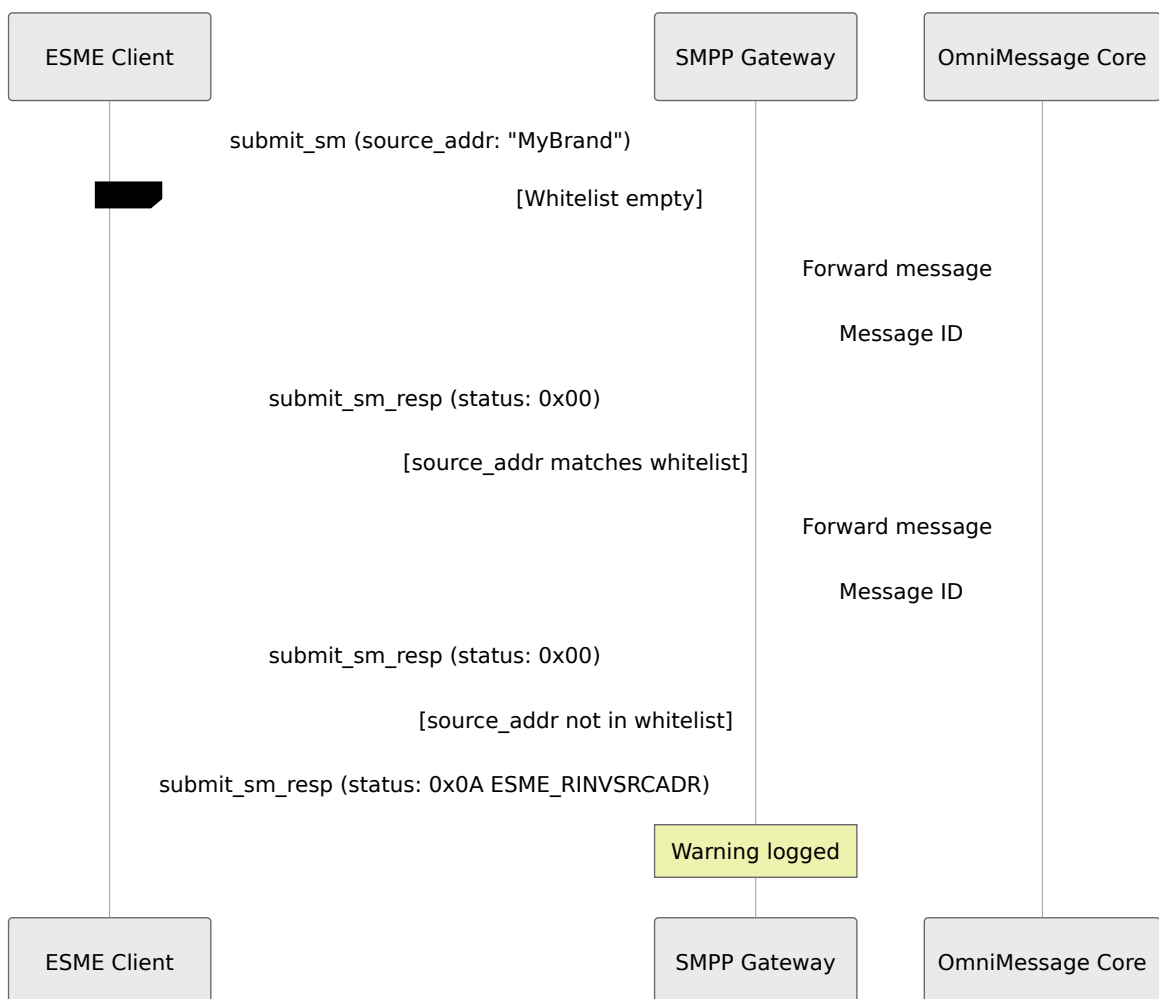
11

変数	型	初期値	参照	説明
source_address_whitelist	リスト	空	参照	許可された送信元アドレスのリスト。このリストに指定されたアドレスからの接続は許可されます。このリストは、このモジュールの初期化時に設定されます。

11

[illegible]

--	--	--	--



□□

□□□□□□□□

□□□□□□ID□AcmeCorp□□□□

AcmeCorp

□□□□□□□□□□

□□□□□□□□□□□□□□614□□□□

614*

□□□□□□□□□□

□□□□□□□□□□□□□□

AcmeCorp,614*,+612900000001

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□**ESME_RINVSRCADR**

□□□□□□□□□submit_sm_resp□□□□□□□0x0A□

□□□□□□

- □□□□□□□□□□□□□□□□□□□□

- 簡易型SMS送信機
- 簡易型SMS受信機
- 簡易型SMS送信機

概要

1. 簡易Web UI
2. 簡易SMS送信機
3. 簡易SMS受信機
4. 簡易SMS送信機

機能

簡易型SMS送信機

仕様

- 簡易型SMS送信機
- ESME
- 簡易型SMS送信機

概要

1. 簡易Web UI
2. 簡易ESME
3. 簡易SMS送信機

機能

- 簡易型SMS送信機
- 簡易型SMS送信機
- 簡易型SMS送信機



XXXXXXXXXX

OmniMessage XXXX

XX SMPP XXXXXXXXXXXXXXXX OmniMessage CoreXXXX OmniMessage XXXXXXXXXXXXXXXX
XXXX

OmniMessage XXXXXXXX

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXAPI XXXXXXXX
- XXXXXX“XXXXXX”“XX”“HTTP 503”“XXXXX”

XX

1. XX OmniMessage XXXX

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

# XXXXXXXXXXXX
ssh gateway-server 'curl -k https://omnimessage-
core.example.com:8443/api/system/health'
```

2. XXXXXX API URL

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

# 確認
ping omnimessage-core.example.com
nc -zv omnimessage-core.example.com 8443
```

3. 確認 API 接続

```
# 確認 API 接続
sudo journalctl -u omnimessage-smpp -f | grep -i
'api\|omnimessage\|connect'

# 確認エラー
sudo journalctl -u omnimessage-smpp -n 200 | grep -i error
```

確認

OmniMessage 確認

- OmniMessage 確認
- 確認
- 確認 `SMPP_POLL_INTERVAL`
- OmniMessage 確認

OmniMessage 確認

- 確認 HTTPS
- DNS 確認 `nslookup omnimessage-core.example.com`
- 確認 `traceroute omnimessage-core.example.com`
- SSL 確認 HTTPS

API URL 確認

- `/opt/omnimessage-smpp/config/runtime.exs`
- `api_base_url` 確認 HTTPS
- `sudo systemctl restart omnimessage-smpp`

目標

前提条件

環境

- 仮想マシン“Ubuntu”を起動
- 仮想マシンをインターネットに接続
- 仮想マシンにSSH接続

確認事項

1. ネットワーク

確認

```
# IP DNS 確認
nslookup smpp.carrier.com

# ping 確認
ping -c 3 smpp.carrier.com

# telnet 確認
telnet smpp.carrier.com 2775
# 接続
nc -zv smpp.carrier.com 2775
```

確認事項

- IP DNS 確認結果から IP アドレスを確認
- ping 確認結果から接続を確認
- telnet 確認結果から接続を確認

2. 接続

確認

- 直接“连接”“数据库”
- Web UI → SMPP → 获取 system_id 信息

数据库

- 数据库连接
- 数据库表结构
- 数据库操作

3. IP 地址

IP

- 公网 IP
- 公网 IP 地址

公网 IP

- 公网 IP 地址

```
curl ifconfig.me
```

- 公网 IP 地址
- 公网 IP 地址

4. 防火墙

防火墙

```
# 检查防火墙
sudo iptables -L -n | grep 2775

# 检查 UFW (Ubuntu/Debian)
sudo ufw status | grep 2775

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

防火墙

```
# Ubuntu/Debian
sudo ufw allow out 2775/tcp

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

故障排除

检查

- 检查网络配置
- `smpp_disconnection_total` 计数器
- 检查日志

网络配置

1. 检查网络

检查

```
# 检查网络
ping -c 100 smpp.carrier.com | grep loss

# 检查网络
netstat -s | grep -i error
```

检查

- 检查网络配置
- 检查网络接口 ISP
- 检查网络/接口

2. 检查网络

检查

- 设置“enquire_link timeout”
- 设置超时时间

设置

- 设置 30 s
- 设置超时时间
- 设置超时时间

3. TPS 设置

设置

- 设置
- 设置

设置

- 设置 `tps_limit` 值
- 设置 TPS 限制为 70-80%
- 设置

4. 设置

设置

- 设置
- 设置

设置

- 设置
 - 设置
-

概要

目的

目的

- 目的
- `smpp_messages_sent_total` 目的
- 目的

目的

1. `dest_smsc` 目的

目的

- Web UI → 目的 → `dest_smsc` 目的
- `SMPP` → 目的

目的

- `dest_smsc` 目的
- `dest_smsc`
- `dest_smsc` `NULL`目的

2. 目的

目的

- Web UI → 目的 → `deliver_after` 目的
- 目的

目的

- `deliver_after`
- 目的

目的

- 配置
- 验证配置

3. TPS 配置

配置

- 配置
- 验证配置

配置

- 配置 `tps_limit`
- 验证配置
- 查看 [CONFIGURATION.md](#)

4. 验证配置

配置

- 配置
- 验证配置

配置

```
# 配置
sudo systemctl restart omnimessage-smpp

# 验证配置
sudo journalctl -u omnimessage-smpp -f
```

配置

配置

- `smpp_delivery_failures_total` 配置
- 配置“submit_sm_resp”配置

- 00000000

0000000000

1. 00000000

000

- 0000000000
- 0000000000

00000000

- 0x0000000B - 0000
- 0x00000001 - 000000
- 0x00000003 - 0000

000000

- 00000000 E.164
- 000000000000
- 00000000

2. 00000000

000

- 0000
- 0000
- 00

000000

- GSM-7000 160 000
- UCS-2000 70 000
- 0000000000
- 00000000

3. 000000

API

- 认证和授权
- 数据格式

API

- 认证和授权
- 数据格式
- 错误处理/响应

4. 部署

API

- 设置 `expires` 头
- 部署

API

- 部署
- 测试和验证

Web UI

部署 Web UI

API

- 部署到 <https://your-server:8087>
- 部署

部署和验证

1. 部署

API

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

□□□□□

```
# □□□□□□□□  
sudo systemctl start omnimessage-smpp  
  
# □□□□□□□□  
sudo journalctl -u omnimessage-smpp -n 50
```

2. □□□□□ 8087 □□

□□□

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

□□□□□

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

3. SSL □□□□

□□□

- □□□□□□□□□□
- □□□□□□□□

□□□□□

- □□□□□□□□□□□□□□□□

- 刪除 SSL 檔案
- 刪除私鑰檔案

```
ls -l /opt/omnimessage-smpp/priv/cert/
```

4. 設定 URL

設定

- 設定 HTTPS 或 HTTP
- 設定伺服器 IP/埠
- 設定 8087

Web UI 安裝

安裝

- 安裝伺服器
- 安裝客戶端
- 安裝

安裝

1. 安裝伺服器

- Ctrl+F5 重新整理
- 清除 Cookie

2. 安裝客戶端

- 按 F12
- 清除 JavaScript 檔案
- 清除   檔案

3. 安裝

- 按 Chrome 或 Firefox 或 Edge 安裝
- 安裝

4. 確認

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

確認

Prometheus 確認

確認

- `curl http://localhost:4000/metrics` 実行
- Prometheus 確認
- 確認

確認

1. 確認

確認

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

確認

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

2. 確認

確認

```
# 本地测试
curl http://localhost:4000/metrics

# 远程测试
curl http://your-server-ip:4000/metrics
```

验证步骤

- 使用 curl 命令测试接口
- 确保 Prometheus 端口 4000 开放

3. 部署

配置

- 添加 `/metrics` 接口 `/prometheus` 和 `/stats`
- 设置 4000 端口和 8087 端口

部署步骤

配置

- 安装 Prometheus
- 安装 Grafana
- 配置数据源

验证

1. 安装 Prometheus

- 下载 Prometheus
- 解压并安装
- 配置 Prometheus `prometheus.yml` 文件 `increase()` 和 `rate()`

2. 安装 Grafana

- 安装 Grafana

- 消息队列
- 数据库

3. 性能优化

- 数据库索引
- 数据库连接池
- 使用 Prometheus 监控系统

性能优化

CPU 使用

查看

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

优化方法

- 减少日志
- 优化 SQL
- 优化配置

监控

- 使用 Prometheus 监控系统
- 设置 TPS 阈值
- 设置 CPU 使用率阈值

内存使用

查看

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

測試環境

- 測試環境配置
- 測試數據生成

測試結果

- 測試結果分析
- 測試結果總結
- 測試結果建議

測試總結

測試

- 測試環境配置
- 測試數據
- 測試結果

測試

1. TPS 測試 - 測試環境
2. `queue_check_frequency` - 測試數據
3. 測試 API 測試 - 測試結果
4. 測試結果分析

測試結果

- 測試結果分析 TPS
 - 測試 `queue_check_frequency` 測試數據
 - 測試 API
 - 測試結果
-

環境構築

インストール

前提

- インストール環境
- インストールコマンド

手順

```
# Elixir インストール
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

確認

- インストール確認
- インストールコマンド vs 'コマンド'
- インストール環境
- インストール `import Config`

確認

- インストール
- インストール
- インストール Elixir インストール環境

インストール

前提

- インストール環境
- インストールコマンド

確認

```
# 重启服务
sudo systemctl restart omnimessage-smpp

# 查看服务状态
sudo systemctl status omnimessage-smpp

# 查看日志
sudo journalctl -u omnimessage-smpp -n 50
```

检查

磁盘空间

内存

1. 检查磁盘空间

```
# 查看磁盘空间
df -h

# 查看内存
free -h

# CPU 使用率
uptime
```

2. 检查服务状态

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

3. 查看日志

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

4. 確認

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

5. 確認

- 確認
- SSL 確認
- 確認
- 確認

6. 確認

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

# 確認
sudo systemctl restart omnimessage-smpp
```

7. 確認

確認

確認

確認

1. 確認 `cat /opt/omnimessage-smpp/VERSION`

2. 確認

```
sudo journalctl -u omnimessage-smpp -n 200 > /tmp/smpp-logs.txt
```

3. 確認

```
sudo cp /opt/omnimessage-smpp/config/runtime.exs  
/tmp/config.exs  
# [] /tmp/config.exs []
```

4. []

```
curl http://localhost:4000/metrics > /tmp/metrics.txt
```

5. []

```
uname -a > /tmp/system-info.txt  
free -h >> /tmp/system-info.txt  
df -h >> /tmp/system-info.txt
```

[]

- [] support@omnitouch.com
- [] +61 XXXX XXXX 24/7
- [] []

[]

- **USAGE.md** - []
- **CONFIGURATION.md** - []
- **MONITORING.md** - []
- **README.md** - []



XXXXXX

XXXXXX OmniMessage Core

XXXX OmniMessage SMPP XXXX OmniMessage Core XXXX
OmniMessage XXXX—XXXX

XXXX OmniMessage XXXX

- XXXX
- XXXX
- XXXX
- XXXX

XXXX **OmniMessage** XXXX

```
# XXX API  
curl -k https://omnimessage-  
core.example.com:8443/api/system/health  
  
# XXXX API URL  
grep api_base_url /opt/omnimessage-smpp/config/runtime.exs
```



XXXXXX

XXXXXXXXXX

1. XXX **Web** XXX

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

- 配置数据库连接

2. 配置

- 配置SMPP → 配置
- 配置“”
- 配置

3. 配置

- 配置
- 配置
- 配置

4. 配置

- 配置
- 配置
- 配置

5. 配置Prometheus

- `curl http://localhost:4000/metrics`
- 配置Grafana
- 配置

--	--	--	--

--	--	--	--	--	--	--	--

- $\text{CO}_2 > 2\text{CO}$
- $\text{CO}_2 > 5\%$
- CO_2
- CO_2

MONITORING.md

--	--	--	--	--	--

OmniMessage Core-SMPP

- **dest_smsc** — `OmniMessage` **dest_smsc:**
"vodafone_uk" `vodafone_uk` `SMPP submit_sm`
- **source_smsc** — `OmniMessage` **source_smsc:**
"partner_acme" `SMPP deliver_sm` `partner_acme`

submit_sm PDU ESME deliver_sm PDU ESME SMSC

□ □ □ □

OmniMessage Core SMPP

- `smsc_name` `"smpp_gateway"` `SMSC_NAME`
- `60`
- `90`
- `{hostname} {peer_name}`

OmniMessage Core

インストールと設定

1. フロントエンド“frontend_register”
 2. `smc_name` OmniMessage
 3. OmniMessage Core `api_base_url`
-

SMPP

SMPP

SMPP

1 Web UI

-
- SMPP → /
-
- Mnesia
-

2

-
- `/opt/omnmessage-smpp/config/runtime.exs`
- Elixir
-
-
-

Web UI

CONFIGURATION.md

環境構築

ESME環境構築

環境構築

- SMPP環境構築/IP
- 2775
- ID
-
-
- TPS

環境構築

A Web UI

環境構築

環境構築

1. 環境構築

- Web UI `https://your-server:8087`
- SMPP →

2. 環境構築

- “”
- - `vodafone_uk`
 - `smpp.vodafone.co.uk`
 - `2775`
 - ID `your_username`
 - `your_password`
 - Transceiver
 - TPS `100`
 - `1000`

- 00“00”

3. 00000000

- 0000000000
- 0000SMPP → 0000
- 000010-3000000“0000”0000
- 000000000000000000

4. 00000000

- 0000000000
- 00000000 dest_smsc 00000000
- 000000000000
- 00000000

00**B**00000000

000000000000000000

000

1. 00000000

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

2. 設定ファイルの編集

```
config :omnimessage_smpp, :binds, [  
  # ...  
  
  # ...  
  %{  
    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  
  }  
]
```

3. サービスの再起動

```
# ... nano ... Ctrl+X ... Y ...  
  
# ...  
sudo systemctl restart omnimessage-smpp
```

4. 確認

- ... SMPP → ...
- ...
- ... “...” ...
- ...

5. 送信

- ...
- ... dest_smsc ...
- ...
- ...

Configuration

Configure the **ESME** and **SMSC** parameters

Steps

1. ESME

- ESME ID `partner_name`
- ESME Name
- ESME Password

2. SMSC

- SMSC IP
- SMSC TPS
- SMSC Port

Configuration

Web UI

Configuration

Steps

1. Web UI

- Web UI `https://your-server:8087`
- SMPP → ESME

2. ESME

- ESME ID `partner_acme`
- ESME Name
 - ESME ID `acme_corp`
 - ESME Password `secure_password_123`
 - ESME Name

- **IP** 203.0.113.0/24
 - **TPS** 50
 - 1000
- “ ”

3.

-
-

4.

- IP
- 2775
- ID acme_corp
- secure_password_123
-

5.

- SMPP →
-
-
- IP

インストールと起動

準備

1. ディレクトリ作成

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

2. 設定ファイルの編集

```
# 接続先server_binds
config :omnimessage_smpp, :server_binds, [
  # 接続先...

  # 接続先設定
  %{
    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
  }
]

# 接続先設定
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

3. サービス再起動

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

4. 確認

- IP
- 2775
- ID acme_corp
- secure_password_123
-

5.

- SMPP →
-
-
- IP

TPS IP

A Web UI

1.

- Web UI <https://your-server:8087>
- SMPP →
- SMPP →

2.

-
- “”
- - TPS IP/
- “”

3. 環境構築

- 環境構築の準備
- 環境構築
- 環境構築SMPP → 環境構築

4. 環境構築

- 環境構築の準備
- 環境構築
- 環境構築

環境構築

環境構築の準備

環境構築

1. 環境構築

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

2. 環境構築

- `binds` : `server_binds` 環境構築
- 環境構築
 - 環境構築TPS環境構築IP環境構築/環境構築
- 環境構築

```
%{  
  name: "vodafone_uk",  
  # ... 環境構築  
  tps_limit: 150, # 環境構築  
  password: "new_password" # 環境構築  
}
```

3. 環境構築

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

4. 設定

- 設定SMPP → 設定
- 設定
- 設定
- 設定

設定

設定SMPP

設定

1. 設定

- 設定/設定
- 設定

2. Web UI

- 設定SMPP → 設定
- 設定
- “設定”
- 設定

3. 設定

- 設定SMPP → 設定/設定
- 設定
- “設定”
- 設定

4. 設定

- 設定 - 設定
- 設定

概要

概要

概要 **enabled** 概要

Web UI

1. 概要SMPP → 概要
2. 概要
3. 概要“概要”
4. 概要“概要”概要
5. 概要“概要”

概要

概要

- 概要
- 概要
- 概要

概要

概要

概要

- 概要30
- 概要500
- 概要

概要

概要SMPP **enquire_link** PDU概要

- 每隔60秒发送enquire_link_interval
- enquire_link_interval: 0
- 每隔60秒发送一次心跳包

Prometheus smpp_enquire_link_sent_total
smpp_enquire_link_received_total

TPS

tps_limit

- 1
- 100
- 100
- 100

1. tps_limit
2. queue_check_frequency
- 3.

1.
 -
 -

2. 送信処理

- 送信機
- 送信
 - 送信機
 - 送信機
 - 送信SMSC[dest_smsc]
 - 送信機
 - 送信

3. 受信処理

- 受信機
- 受信機SMSC

送信機

送信機

送信

1. 送信機

- 送信SMPP → 送信機
- 送信機
- 送信機送信機

2. 消息的发送

- 消息的发送
- 消息的接收
- 消息的 `dest_smsc` 消息的接收
- 消息的 `deliver_after` 消息的接收

3. 消息的接收

- 消息 = 消息
- 消息的接收
- 消息  消息的接收

4. 消息的接收

- 消息的接收
- 消息的接收
- 消息的接收

消息的接收

消息的接收

消息的接收“消息”消息

消息

1. 消息的接收

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

2. 消息的接收

- 消息的接收
- 消息的接收

- 0000000000000000

3. 000000

- 0000SMPP → 000/0000000
- 0000ID00000000
- 00000000000000

4. 00000000

- 0000SMPP → 0000
- 00000000
- 00“0000”00
- 0010-300
- 0000000000“000”

5. 0000000000

- 00000000
- 0000000000000000
- 00000000
- 000TROUBLESHOOTING.md

0000000000

00000000000000

目標

- 認証/認可
- IPアドレス管理
- ログ/監視

概要

1. 認証
 - 認証SMPP → 認証
 - 認証ID
 - 認証
2. IP管理
 - IPアドレス
 - IP
3. ログ
 - ログ
 - ログ
 - ログ
4. 監視
 - Web UI
 - “”

環境

Prometheus

環境

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

🔍 🔍 🔍 🔍

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

🔍 🔍 🔍 🔍 1 🔍 🔍 🔍 🔍

🔍 🔍 🔍 🔍

🔍 🔍 🔍 🔍 🔍 🔍

1. 🔍 🔍 Web UI → SMPP → 🔍 🔍 🔍
2. 🔍 🔍 🔍 🔍 🔍 🔍
3. 🔍 🔍 🔍 🔍 🔍 🔍
4. 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍
5. 🔍 🔍 [TROUBLESHOOTING.md](#)

🔍 🔍 🔍 🔍 🔍 🔍

1. 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍
2. 🔍 🔍 🔍 🔍 🔍 🔍 🔍
3. 🔍 🔍 🔍 🔍 🔍 🔍 🔍
4. 🔍 🔍 🔍 🔍 🔍 🔍 🔍

🔍 🔍 🔍 🔍 🔍

1. 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍 🔍
 2. 🔍 🔍 `dest_smsc` 🔍 🔍 🔍 🔍 🔍
 3. 🔍 🔍 TPS 🔍 🔍 🔍 🔍 🔍 🔍
 4. 🔍 🔍 `queue_check_frequency` 🔍 🔍
-

目錄

目錄

目錄

1. 目錄

- 目錄
- 目錄
- 目錄

2. 目錄

- 目錄
- 目錄
- 目錄

3. 目錄

- 目錄SMPP
- 目錄
- 目錄IP

4. 目錄

- 目錄
- 目錄TPS
- 目錄

目錄

目錄

- 目錄
- 目錄
- 目錄

000

```
# 000000
sudo systemctl status omnimessage-smpp

# 0000
sudo systemctl restart omnimessage-smpp

# 0000
sudo systemctl status omnimessage-smpp

# 0000
sudo journalctl -u omnimessage-smpp -n 50
```

00**Web UI**000

1. 000000000030-60000000
2. 0000SMPP → 0000
3. 00000000001-2000
4. 0000000000

00000

000000000000💎💎

```
# 0000
sudo cp /opt/omnimessage-smpp/config/runtime.exs \
  /opt/omnimessage-smpp/config/runtime.exs.backup.$(date +%Y%m%d)

# 0000
sudo tar -czf /tmp/smpp-certs-$(date +%Y%m%d).tar.gz \
  /opt/omnimessage-smpp/priv/cert/
```

00000000

```
# 確認
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup.YYYMMDD \
/opt/omnimessage-smpp/config/runtime.exs

# 再起動
sudo systemctl restart omnimessage-smpp
```

確認

再起動確認

確認

1. サービス確認

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

2. サービス再起動

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

3. ログ確認

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

4. 環境確認

- サービス再起動
- SSL証明書確認
- ディスク容量 `df -h`
- メモリ使用量 `free -h`

5. サービス再起動

TPS 101

TPS

1. TPS 101

- TPS 101 → TPS 101
- TPS 101
- TPS “TPS 101”

2. TPS 101

- TPS 101
- TPS 101
- TPS 101

3. TPS 101

- TPS 101
- TPS 101
- TPS 101

4. TPS 101

- TPS 101
- TPS 101
- TPS 101

TPS 101

TPS 101

TPS

1. TPS 101

- TPS 101 → TPS 101
- TPS 101
- TPS 101 TPS 101

2. 認證與授權

- 認證與授權
- 認證與授權

3. 系統架構

- 系統架構
- 系統架構

4. 系統部署

- 系統部署
- 系統部署

系統部署

系統部署

- ☐ 系統部署SMPP系統
- ☐ 系統部署系統
- ☐ 系統部署系統
- ☐ 系統Prometheus/Grafana系統
- ☐ 系統部署 > 98%

系統部署

- ☐ 系統部署
- ☐ 系統部署
- ☐ 系統部署
- ☐ 系統部署
- ☐ 系統部署

目录

- ☐ 简介
 - ☐ 快速入门
 - ☐ 部署指南
 - ☐ 配置TPS
 - ☐ 故障排除
-

目录

- **CONFIGURATION.md** - 配置指南
 - **SOURCE_ADDRESS_WHITELIST.md** - 源地址白名单
 - **MONITORING.md** - Prometheus监控
 - **TROUBLESHOOTING.md** - 故障排除
 - **README.md** - 简介
-

