


```

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	配置 (URL)	配置	-

配置 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

- ✓ `api_base_url`
- ✓ `api_base_url`
- ✓ `api_base_url`
- ✓ `api_base_url`
- ✓ `api_base_url`
- ✓ `api_base_url`
- ✓ `api_base_url`

Configure SMPP ↔ OmniMessage

Example

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

# OmniMessage Core over HTTPS
api_base_url: "https://omnimessage-core.company.com:8443"

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

Notes

- Configure OmniMessage Core
- Configure HTTPS `verify_ssl_peer`
- Configure `verify_ssl_peer` HTTPS

SMPP_POLL_INTERVAL

Example

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

項目	単位	範囲	既定値
smpp_poll_interval	ms	100-2000	100

このパラメータは、SMPPメッセージの送信間隔を制御します。

調整方法

- TPS > 100: 100-500ms
- TPS 10-100: 500-1000ms
- TPS < 10: 1000-2000ms

設定コマンド: `SMPP_POLL_INTERVAL`

VERIFY_SSL_PEER

SSL 接続

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

項目	単位	範囲	既定値
verify_ssl_peer	boolean	true/false	false

このパラメータは、API 接続時の SSL 接続を制御します。

調整方法

- true: SSL 接続を有効にする
- false: SSL 接続を無効にする

設定コマンド: `VERIFY_SSL_PEER`

□□□□□□□□□□

```
config :omnimessage_smpp, :binds, [  
  %{  
    # □□□□□□□□□□  
    name: "vodafone_uk",  
  
    # □□□□  
    mode: :client,  
  
    # SMPP □□□□  
    bind_type: :transceiver,  
  
    # □□□ SMPP □□□□□□  
    host: "smpp.vodafone.co.uk",  
    port: 2775,  
  
    # □□□□□□  
    system_id: "your_username",  
    password: "your_password",  
  
    # SMPP □□□□□□□□□□□□□□□□□□□□  
    system_type: "",  
    addr_ton: 0,  
    addr_npi: 0,  
    address_range: "",  
  
    # □□□□  
    tps_limit: 100,  
  
    # □□□□□□  
    queue_check_frequency: 1000,  
  
    # □□□□□□□□□□□□□□□□□□□□  
    enquire_link_interval: 60,  
  
    # □□□□□□□□□□  
    cache_enabled: false,  
    cache_max_size: 10000,  
    cache_retry_interval: 60  
  }  
]
```

□□□□□□□□

name

□□□□□□□□

□□	□□	□□
□□□	□	"vodafone_uk"

□□□□□□□□ SMPP □□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□

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

mode

□□□□

□□	□□	□
□□	□	:client

□□□□□□□□□□□□□□□□ **ESME** □□□□□ **SMSC**□

□□□□□□□□□□□□ □:client□

bind_type

SMPP □□□□

名前	タイプ	値
transmitter	string	:transmitter
receiver	string	:receiver
transceiver	string	:transceiver

送信機と受信機

送信機

- `:transmitter` - `submit_sm` を送信する
- `:receiver` - `deliver_sm` を受信する
- `:transceiver` - 送信機と受信機を兼ねる

送信機と受信機を兼ねる `:transceiver`

host

送信機 SMPP 送信機 IP

名前	タイプ	値
host	string	"smpv.carrier.com" 10.5.1.100

送信機 SMPP 送信機

送信機

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

port

送信機 SMPP 送信機

名前	タイプ	値	説明
port	integer	2775	1-65535

SMPP TCP

2775

```
port: 2775 #  
port: 3000 #
```

system_id

		"company_user"

-

password

		"secret_password"

-
-
-

tps_limit

队列大小

配置	默认	范围	单位
队列大小	100	1-10000	

队列大小对性能的影响

队列大小

- 队列大小对性能影响 70-80%
- 队列大小/队列大小
- 队列大小对性能影响

队列大小

```
tps_limit: 10 # 队列大小  
tps_limit: 50 # 队列大小  
tps_limit: 100 # 队列大小  
tps_limit: 1000 # 队列大小
```

队列大小

```
队列大小 = 100 TPS  
队列 tps_limit = 70-80  
队列 20-30 TPS 队列
```

queue_check_frequency

队列检查频率

配置	默认	范围	单位
队列检查频率	1000	100-10000	

队列检查频率对性能的影响

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

address_range

address_range

address_range

name	type	value	default
address_range	string	"614*"	

address_range SMPP address_range

enabled

enabled

name	type	value	default
enabled	boolean	true	

enabled   enabled

cache_enabled

cache_enabled

name	type	value	default
cache_enabled	boolean	false	

cache_enabled API cache_enabled

[MESSAGE_CACHE.md](#)

cache_max_size

SMPP 配置

配置 SMSC 和 ESME 参数

配置示例

```
config :omnimessage_smpp, :server_binds, [  
  %{  
    # 名称  
    name: "partner_acme",  
  
    # 系统 ID 和密码  
    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  
    source_address_whitelist: [],  
  
    # TPS 限制  
    tps_limit: 50,  
  
    # 队列检查频率  
    queue_check_frequency: 1000,  
  
    # 询问链路间隔  
    enquire_link_interval: 60,  
  
    # 缓存配置  
    cache_enabled: false,  
    cache_max_size: 10000,  
    cache_retry_interval: 60  
  }  
]
```

□□□□□□□□

name

□□□□□□

□□	□□	□□
□□□	□	"partner_acme"

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□/□□□□□□□□□□□□

system_id

□□□□□□□□

□□	□□	□□
□□□	□	"acme_corp"

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

password

□□□□□□□□

□□	□□	□□
□□□	□	"secure_password"

□□□□□□□□□□□□□□□□□□□□□□

□□□□

- 00000
- 00000000
- 000000000000

allowed_bind_types

00000000

00	00	00
00000	0	-

000000000000000000000000

000

```
allowed_bind_types: [:transceiver] # 0 transceiver
allowed_bind_types: [:transmitter, :receiver] # TX 0 RX
allowed_bind_types: [:transmitter, :receiver, :transceiver] # 00
```

000000000000000000000000

ip_whitelist

000000 IP 00

00	00	00	00
000000	0	[]	IP 0 CIDR 000

000000 - 00000000 IP 00000

000

- 00 IP "192.168.1.100" 000 /32
- CIDR 000 "192.168.1.0/24" 0 "10.0.0.0/8"
- 000 ["192.168.1.0/24", "10.50.1.100"]

□□□

```
# □□□□ IP□□□□□□
ip_whitelist: []

# □□ IP
ip_whitelist: ["203.0.113.50"]

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

# □□
ip_whitelist: ["192.168.1.0/24"]

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

□□□□□

- /32 - □□ IP□□□□□□□□ IP □□□
- /24 - 256 □□□□□□□□192.168.1.0-255□
- /16 - 65,536 □□□□□□□□10.50.0.0-255.255□
- /8 - 16,777,216 □□□□□□□□10.0.0.0-255.255.255.255□

source_address_whitelist

□□□□□□

□□	□□	□□	□□
□□□□□□	□	[]	□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ ID□□□□□□□□□□□□□□

□□□□□

- □□□□□□ "MyBrand" □□□ "MyBrand"
- □□□□□□ "614*" □□□□□ "614" □□□□□

配置

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

主机

host

配置 IP

配置	值	说明	备注
host	["0.0.0.0"]	监听地址	["0.0.0.0"] 或 ["127.0.0.1"]

配置 IP 列表

配置

- "0.0.0.0" - 监听所有 IP
- "127.0.0.1" - 监听本地回环地址
- "192.168.1.10" - 指定 IP 地址

port

配置 TCP

配置	值	说明	备注
port	2775	监听端口	1-65535

配置 SMPP 配置

配置 2775

max_connections

□□□□□□□

□□	□□	□□	□□
□□	□	100	1-10000

□□□□□□□□□□□□□□□□

□□□

- □□□□□□□□□□
 - □□□□□□□□□
 - □□□10-100 □□□
-

□□□□□□

□□ **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
  }
]
```

00 20000000

```
import Config

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

config :omnimessage_smpp, :binds, [
  # 00
  %{
    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
  },

  # 00
  %{
    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
  }
]
```

00 3000000000000000

```
import Config

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

编译

编译并安装

安装

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

运行并查看日志

安装

```
# 安装 systemd 服务
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-smpp console
```

按 **Ctrl+C** 退出

编译

编译并安装 systemd 服务

環境変数	設定名	設定値
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"
```

□□□□□□

1. □□□□□□

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

2. □□□□□□

- □□ 12 □□□
- □□□□□□□□□□
- □□□□□□

3. □□ **IP** □□□□

- □□□□□□□□□□ `ip_whitelist`
- □□□□□□□□□□□□□□ `[]`

4. □□ **SSL** □□□

- □□ `verify_ssl_peer: true` □□□□□□□□

5. □□□□□□□

- □□□□□□□□
- □□□□□/□□□□□□□□

□□□

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



□□□□

A

API □□□□□□□□

□□□□□□□□□□□□□□□□

□□□□

Web UI □□□□□□□□□□□□□□□□□□□□

B

□□

SMPP □□□□□□□□□□□□□□□□□□

□□

□□□□□□ SMPP □□□□□□□□□□□□□□□□

□□□□

SMPP □□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□

SMPP □□□□□□□□□□□□□□□□ IP □□□

C

CIDR □□□□□□□□

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

E

ESME

ESME SMPP

ESM

ESM SMPP

ESME

ESME SMPP SMSC ESME SMSC ESME

ESME

ESME 1 2 4 8

F

F

F

F

F SMPP OmniMessage Core 60 90 OmniMessage

G

G

G SMPP

G

G

Grafana

Grafana Prometheus

GSM-7

GSM-7 SMS 7 160

H

HTTP/HTTPS

Web 与 HTTPS 的区别

I

IP 地址

IP 地址的组成与分类

ISDN 接口

ISDN 接口的应用

J

()

K

enquire_link

enquire_link 的应用

KPI 指标

KPI 指标的监控

L

Prometheus

Prometheus 的 bind_name="vodafone_uk"

Phoenix

Web UI 与 Phoenix 的集成

M

消息

消息的发送和接收

消息

消息的发送和接收 Prometheus 消息

MO消息

消息的发送和接收

MT消息

消息的发送和接收

MSISDN消息

消息的发送和接收

N

NPI消息

消息的发送和接收 SMPP 消息 ISDN 消息

O

消息

消息的发送和接收

消息

消息的发送和接收

P

PDU消息

消息 SMPP 消息 submit_sm deliver_sm

□□□

□□□□□□ 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	□□□□□□□□

□□□	□□
SMS	□□□□□
SSL	□□□□□□
TLS	□□□□□
TON	□□□□
TPS	□□□□□
UCS	□□□□□□□
UI	□□□□
URL	□□□□□□□

□□□□

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

SMPP 架构图

简介

SMPP 系统架构图展示了 SMPP 客户端 API 与 Mnesia 数据库以及 Backend API 的交互流程。

组件

- SMPP 客户端 - 调用 API 提交 SMPP 消息
- Mnesia 数据库 - 存储消息并支持 `:disc_copies` 配置
- 消息队列 - 用于异步处理消息
- Gateway Server - 接收并转发 SMPP 消息
- 消息队列 - 实现 FIFO 队列
- 消息队列 - 用于异步处理消息
- 消息队列 - 支持 LiveView 实时监控
- **Prometheus** 监控 - 用于系统性能监控

流程

提交消息

消息处理

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,
  # 配置缓存刷新间隔
  cache_flush_interval: 10_000,

  # 配置失败永久重试次数
  cache_max_retry_attempts: 10,

  # 配置缓存回溯乘数
  cache_backoff_multiplier: 2,

  # Mnesia 存储类型 (: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",

    # 配置缓存
    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

```

# 配置缓存刷新间隔
CACHE_FLUSH_INTERVAL=10000           # 是否启用缓存
# 配置失败永久重试次数
CACHE_MAX_RETRY_ATTEMPTS=10          # 缓存最大大小
# 配置缓存回溯乘数
CACHE_BACKOFF_MULTIPLIER=2           # 缓存重试间隔
# Mnesia 存储类型
MNESIA_STORAGE_TYPE=disc_copies

```

□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

```

□□□□□60□
□□□□□2

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

```

□□□□

□ 10 □□□□□□□□□□□□□□□□ failed_permanent □□□

- □□□□□□□□□□□□
- □□□❓❓□□
- □□□□□□□□□□“□□□□”□□
- □□□□□□□□□□

□□□□

```

:pending → :delivering → SUCCESS (□□□□□□)
           → FAILURE → :pending (□□□□□□)
           → :failed_permanent (□□□□□□)

```



LiveView

`http://your-server:4000/smpp` → “”

- -
- -
- -

-
-
- /
-
-

Prometheus

```
#   
smpp_cache_size{bind_name="my_bind",mode="server"} 42  
  
#   
smpp_cache_delivered_total{bind_name="my_bind"} 1234  
  
#   
smpp_cache_retry_total{bind_name="my_bind"} 56  
  
#   
smpp_cache_permanent_failures_total{bind_name="my_bind"} 2  
  
#   
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** □□□

```

# 更新绑定配置
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: true,
  cache_max_size: 10_000,
  cache_retry_interval: 60
)

# 禁用缓存
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: false
)

```

查看缓存

```

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

# 按绑定查看缓存
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 消息状态改为 pending
SmsC.SMPP.MessageCache.update_cache_record("my_bind", -123456, %{
  status: :pending,
  retry_count: 0,
  next_retry_at: DateTime.utc_now(),
  last_error: nil
})

```

配置

配置 / 配置

配置 `cache_overflow_total` 配置

配置

配置

1. 配置 `cache_max_size`
2. 配置 API 配置 API 配置
3. 配置
4. 配置

□□□□□□

□□□ □□□□□ □□ :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 □

□□□

```

# 消息队列
{: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)

# 打印 last_error
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 简介

消息队列

简介 Mnesia 简介

```

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

```

简介

1. 消息队列
2. 消息队列 `cache_max_size`
3. 消息队列 FIFO 简介

메시지

메시지

- 메시지의 수 500-1000 메시지에 제한
- 10,000 메시지 ≈ 5-10 MB
- `:disc_copies` 메시지 수

CPU

- 메시지 수 10 메시지에 제한
- 메시지 수 100 메시지에 제한
- 메시지에 제한 10 메시지에 API

메시지 I/O

- `:disc_copies` 메시지 수
- 메시지에 제한 >1000 msg/sec 메시지에
 - `:ram_copies` 메시지 수
 - 메시지 수
 - 메시지 수

메시지

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

□□□□

□□□□□□


1. Mnesia □□□□□□ □ `:disc_copies` □
2. □□□□□□□□
3. □□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□

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

API □□□□

1. □□□□□□□□□□□□
2. □ API □□□□□□□□□□□□□□
3. □□□□□□□□□□FIFO□
4. □□□□□□□□□□ API □□

□□□□

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



- □□□□
- □□□□□
- □□□□

SMPP Metrics

SMPP Overview

Intro

SMPP metrics are exposed via Prometheus endpoint

OmniMessage Core **OmniMessage** metrics are exposed via

1. **SMPP** metrics - `/smpp`
2. **OmniMessage API** metrics - `/omni`

Example

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

Example Prometheus query

`smpp_omni_message_core`

CLI

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

Labels

Labels include `smpp_` prefix


```
- alert: SMPP_License_Invalid
  expr: omnimessage_smpp_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "SMPP"
    description: " - "
```

smpp_connection_status

Gauge

SMPP

- 1 =
- 0 =

- bind_name - "vodafone_uk"
- mode - "client" "server"
- host -
- port -
- bind_type - SMPP
- system_id - ID

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

- 0000000000000000
- 0000000000000000
- 00000000

000000

smpp_messages_sent_total

Counter

SMPP0000000000

0000

connection_status

000

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

000

- 00000000/00
- 0000/000
- 0000000000

smpp_messages_received_total

Counter

SMPP0000000000

0000

connection_status

000

```
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
```

□□□□□□

smpp_bind_success_total

□□□Counter

□□□□□□□□□□□□

□□□□□□□

□□□

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

□□□

- □□□□□□□
- □□□□□□□

smpp_bind_failures_total

□□□Counter

□□□□□□□□□□□□

□□□□□□□

□□□

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

□□□

- □□□□□□□□□□
- □□□□□□
- □□□□□□□□

□□□□□□

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

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

- > 0   

smpp_cache_overflow_total

Counter

000

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

000

- 00000000000000000000API00000000

OmniMessage API

0000000000SMPP00000000OmniMessage API0000000000000000

00OmniMessage000000000000

- omnimessage_api_requests_total - 00000000API00000
- omnimessage_api_request_duration_seconds - API00000
- omnimessage_queue_depth - OmniMessage00000000000

00000000000000000000

0000000000API0000

- "api.*connection refused" - 000000OmniMessage
- "api.*timeout" - OmniMessage0000
- "api.*http 503" - OmniMessage000000
- "api.*parse error" - 00000000

Prometheus

00000000

0000/etc/prometheus/prometheus.yml

```
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

Queries

Alerts

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

Stat

Alerts

- $value < 1000000$
- $value == 1000000$

Queries

Alerts

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

Graph

Y-axis

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

仪表盘

概览

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

仪表盘 Gauge

仪表盘 0-100%

概览

- 仪表盘 < 95%
- 仪表盘 95-98%
- 仪表盘 > 98%

仪表盘 仪表盘

概览

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

仪表盘 Stat

仪表盘

仪表盘

Prometheus 仪表盘

仪表盘 `/etc/prometheus/rules/smpp-alerts.yml` 仪表盘

```

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

    # 失败率过高
    - 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 }}失败率为{{ $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"

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

```



```
    expr: rate(smpp_messages_sent_total[10m]) == 0 and
rate(smpp_messages_received_total[10m]) == 0
    for: 30m
    labels:
      severity: warning
    annotations:
      summary: "30m SMPP API 消息发送/接收为0"
      description: "30m SMPP API 消息发送/接收为0"
  ""
```

prometheus.yml 配置

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

Web 访问

Web UI 访问 Prometheus



URL: https://your-server:8087

访问

SMPP → 配置

配置

- 配置
- 配置
- 配置
- 配置/配置
- 5配置

配置


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

# 100行確認
sudo journalctl -u omnimessage-smpp -n 100

# 1時間前からの確認
sudo journalctl -u omnimessage-smpp --since "1 hour ago"

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

Web UI

Web UI

-
-
-
-
-

- 平均接続時間
- 送信メッセージ数
- 送信メッセージ数/送信失敗メッセージ数
- 送信/送信失敗メッセージ数
- 送信成功率

送信接続時間 KPI

送信時間

送信接続時間

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

送信時間 > 99.9%

送信メッセージ数

送信メッセージ数

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

送信失敗メッセージ数

送信成功率

送信成功率

```
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
 - 配置
-

配置

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** - □□□□□□□□
-

Brand	Number	Result
MyBrand	MyBrand 1234	Success
61412345678	614* 1234	Success
+61400000001	+61400000001 1234	Success
OtherBrand		Failure
61500000001		Failure

1234

submit_sm 1234

Field	Value
PDU	submit_sm_resp
ESME	0x0000000A
ESME_RINVSRCADR	1234
ESME_ID	

1234

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  
  }  
]
```


□□

□□□□□□□□

□□□□□□ID AcmeCorp□□□□

AcmeCorp

□□□□□□□□□□

□□□□□□□□□□□□□□ 614□□□□

614*

□□□□□□□□□□

□□□□□□□□□□□□□□

AcmeCorp, 614*, +61290000001

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□ **ESME_RINVSRCADR**

□□□□□□□□ submit_sm_resp□□□□□□ 0x0A□

□□□□□

- □□□□□□□□□□□□□□□□□□

- 系統設計
- 系統開發
- 系統測試

系統設計

1. Web UI設計
2. 系統架構設計
3. 資料庫設計
4. 安全設計

系統開發

系統開發

系統測試

- 單元測試
- ESME測試
- 系統測試

系統部署

1. Web UI部署
2. ESME部署
3. 系統部署

系統維護

- 系統監控 - 系統運行狀態
- 系統更新 - SMPP更新
- 系統備份 - 數據備份


```
# 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

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

# エラーを確認
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
- 確認 `sudo systemctl restart omnimessage-smpp`

□□□□

□□□□□□

□□□

- □□□□“□□□□”□□□□
- □□□□□□□□□□
- □□□□□□□□

□□□□□□□□□□

1. □□□□□□

□□□

```
# □□ DNS □□  
nslookup smpp.carrier.com  
  
# □□□□  
ping -c 3 smpp.carrier.com  
  
# □□□□  
telnet smpp.carrier.com 2775  
# □  
nc -zv smpp.carrier.com 2775
```

□□□□□

- □□ DNS □□□□□□□□□□ IP □□□□□□□□□□
- □□ ping □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□

2. □□□□□□

□□□

- 0000“0000”0“000000”
- Web UI0SMPP → 000000 → 00 system_id 000

00000

- 00000000
- 00000000000000
- 0000000000

3. IP 000000

000

- 00000000
- 00000000000000 IP

00000

- 00000000 IP

```
curl ifconfig.me
```

- 000000 IP 000000
- 00 IP 0000000000 IP

4. 00000

000

```
# 0000000000
sudo iptables -L -n | grep 2775

# 00 UFW0Ubuntu/Debian0
sudo ufw status | grep 2775

# 00 firewalld0RHEL/CentOS0
sudo firewall-cmd --list-ports | grep 2775
```

00000

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

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

□□□□□□

□□□

- □□□□□□□□□□
- `smpp_disconnection_total` □□□□
- □□□□□□□□

□□□□□□□□□□

1. □□□□□

□□□

```
# □□□□
ping -c 100 smpp.carrier.com | grep loss

# □□□□□□
netstat -s | grep -i error
```

□□□□□

- □□□□□□□□□□
- □□□□□□□□□ ISP
- □□□□□□□/□□

2. □□□□□□

□□□

- 检查“enquire_link timeout”
- 检查网络配置

检查

- 检查 30 秒
- 检查网络配置
- 检查网络配置

3. TPS 检查

检查

- 检查配置
- 检查配置

检查

- 检查 `tps_limit` 配置
- 检查 TPS 配置 70-80%
- 检查配置

4. 检查

检查

- 检查配置
- 检查配置

检查

- 检查配置
 - 检查配置
-

□□□□□□

□□□□□

□□□

- □□□□□□
- `smpp_messages_sent_total` □□□□
- □□□□□□

□□□□□□□□□□

1. □□□ `dest_smsc` □□

□□□

- Web UI → □□ → □□□□ `dest_smsc` □□
- □ SMPP → □□□□□□□□□□□□□□

□□□□□

- □□□□ `dest_smsc` □□□□
- □□□□□□□□□□□□ `dest_smsc`
- □□ `dest_smsc` □ NULL□□□□□□□□

2. □□□□□□□□□□

□□□

- Web UI → □□ → □□ `deliver_after` □□
- □□□□□□□□□□□□□□

□□□

- □□□□□□□□□□□□ `deliver_after`
- □□□□□□□□□□□□□□

□□□□□

- 000000
- 00000000000000000000

3. TPS 0000

000

- 000000
- 00000000

00000

- 000000 tps_limit
- 00000000000000000000
- 000 [CONFIGURATION.md](#)

4. 0000000000

000

- 0000
- 000000

00000

```
# 0000
sudo systemctl restart omnimessage-smpp

# 0000
sudo journalctl -u omnimessage-smpp -f
```

00000000

000

- smpp_delivery_failures_total 00
- 0000“submit_sm_resp”000000

- 00000000

000000000000

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-7 160 000
- UCS-2 70 000
- 00000000
- 00000000

3. 000000

□□□

- □□□□□□□□□□
- □□□□□□□□

□□□□□

- □□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□/□□□□

4. □□□□

□□□

- □□ expires □□□
- □□□□□□

□□□□□

- □□□□□□□
- □□□□□□□□□□□□

Web UI □□

□□□□ Web □□□

□□□

- □□□□□□□□ <https://your-server:8087>
- □□□□□□□□

□□□□□□□□□□

1. □□□□□

□□□

```
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 □□□□

□□□

- □□□□□□□□□□
- □□□□□□□□

□□□□□

- □□□□□□□□□□□□□□□□

- 証明書 SSL 用
- 証明書生成

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

4. URL 設定

設定

- 証明書 HTTPS 証明書 HTTP
- 証明書 IP/ポート
- 証明書 8087

Web UI 設定

設定

- 証明書
- 証明書
- 証明書

証明書

1. 証明書

- Ctrl+F5 強制更新
- 証明書 Cookie

2. 証明書

- F12
- 証明書 JavaScript 用
- 証明書 🔒 🔒 証明書

3. 証明書

- Chrome Firefox Edge 用
- 証明書

4. 確認

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

確認

Prometheus 確認

確認

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

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

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

•

-
- Prometheus 4000

3. Prometheus

•

- /metrics /prometheus /stats
- 4000 8087

•

•

-
-
-

•

1. Prometheus

-
-
- Prometheus `increase()` `rate()`

2. Prometheus

-

- `ps aux | grep omnimessage-smpp`
- `top -p $(pgrep -f omnimessage-smpp)`

3. `df`

- `df -h`
 - `df -i`
 - `df` Prometheus `node_exporter`
-

📊

CPU 📊

📊

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

📊

- 📊
- 📊
- 📊

📊

- 📊
- 📊 TPS 📊
- 📊 CPU📊

📊

📊

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

□□□□□

- □□□□□□□□
- □□□□□□□□

□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□

□□□□□□

□□□

- □□□□□□□□
- □□□□
- □□□□□

□□□

1. TPS □□ - □□□□□□
2. `queue_check_frequency` - □□□□
3. □□ API □□□□ - □□□□
4. □□□□□□□□□□

□□□□□

- □□□□□□□□□□ TPS
 - □□ `queue_check_frequency` □□□□□□
 - □□□□ API
 - □□□□□□□
-

□□□□

□□□□□□□□

□□□

- □□□□□□□□□□
- □□□□"□□□□"□"□□□□"

□□□

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

□□□□□

- □□□□□□□□□□
- □□□□□□" vs '□'
- □□□□□□□□□□
- □□□□ `import Config`

□□□□□

- □□□□□□
- □□□□□□□
- □□□□ Elixir □□□□□□□□□□

□□□□□□

□□□

- □□□□□□□□□□□□□□
- □□□□□□□□

□□□□□

```
# 重新启动
sudo systemctl restart omnimessage-smpp

# 检查状态
sudo systemctl status omnimessage-smpp

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

检查

磁盘

内存

1. 检查磁盘

```
# 查看磁盘
df -h

# 查看内存
free -h

# CPU 使用率
uptime
```

2. 检查服务

```
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** - []

- 消息队列

2. 消息队列

- 消息队列SMPP → 消息队列
- 消息队列“消息”队列
- 消息队列

3. 消息队列

- 消息队列
- 消息队列
- 消息队列

4. 消息队列

- 消息队列
- 消息队列
- 消息队列

5. 消息队列Prometheus

- `curl http://localhost:4000/metrics`
- 消息队列Grafana
- 消息队列

□□□□□□□□□□□□□□□□□□□□

1. □□□□□□“frontend_register”□□
 2. `smsc_name` □□□□OmniMessage□□□□□□
 3. □□OmniMessage Core□□□□□□□□`api_base_url`□
-

□□SMPP□□

□□□□SMPP□□□□

SMPP□□□□□□□□□□□□□□□□□□□□

□□1□Web UI□□□□□

- □□□□□□□□□□□□□□□□
- □□□□SMPP → □□□□□□□□ / □□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□Mnesia□□□□□□
- □□□□□□□□□□□□□□□□□□

□□2□□□□□□

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

□□□□Web UI□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□`CONFIGURATION.md`□

□□□□□□□□

□□□□□□□□□□□□□□□□ **ESME**□□□□□□

□□□□□□□□□□□□

- SMPP□□□□□□/IP
- □□□□□□□□2775□
- □□ID□□□□□□
- □□
- □□□□□□□□□□□□
- TPS□□

□□□□□□□□□□

□□**A**□□□□**Web UI**□□□□□□

□□□□□□□□□□□□

□□□□

1. □□□□□□□□□□□□

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

2. □□□□□□□□

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

- “”

3.

-
- SMPP →
- 10-30 “”
-

4.

-
- dest_smsc
-
-

B

1.

```
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でCtrl+Xで保存  
  
# サービスの再起動  
sudo systemctl restart omnimessage-smpp
```

4. 確認

- SMPP → ...
- ...
- "..."
- ...

5. 確認

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

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

3.

-
-

4.

- IP
- 2775
- ID acme_corp
- secure_password_123
-

5.

- SMPP →
-
-
- IP

- 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. 設定

- 設定
- 設定
- 設定SMPP → 設定

4. 設定

- 設定
- 設定
- 設定

設定

設定

設定

1. 設定

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

2. 設定

- `server_binds` 設定
- 設定
 - 設定TPS/IP/
- 設定

```
%{  
  name: "vodafone_uk",  
  # ... 設定  
  tps_limit: 150, # 100  
  password: "new_password" # 設定  
}
```

3. 設定

```
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

- 每隔60秒发送enquire_link_interval
- enquire_link_interval: 0
- 每隔60秒发送enquire_link_interval

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. 消息的发送

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

3. 消息的接收

- 消息的接收
- 消息的接收 SMSC

消息的接收

消息的接收

消息


1. 消息的接收

- 消息的接收 SMPP → 消息
- 消息的接收
- 消息的接收 消息

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 → 000/00000000
- 0000ID00000000
- 00000000000000

4. 00000000

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

5. 0000000000

- 00000000
- 0000000000000000
- 00000000
- 000TROUBLESHOOTING.md

0000000000

00000000000000


```
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` □□
-

□□□□

□□□□

□□□□

1. □□□□

- □□□□□□
- □□□□□□
- □□□□□□

2. □□□□

- □□□□□□□□
- □□□□□□
- □□□□□□□□

3. □□□□

- □□□□SMPP□□
- □□□□□□
- □□IP□□□□□□□□

4. □□□□

- □□□□□□□□
- □□TPS□□
- □□□□

□□□□

□□□□

- □□□□□□□□
- □□□□□□
- □□□□□□□□

□□□

```
# □□□□□□
sudo systemctl status omnimessage-smpp

# □□□□
sudo systemctl restart omnimessage-smpp

# □□□□
sudo systemctl status omnimessage-smpp

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

□□Web UI□□□

1. □□□□□□□□□□30-60□□□□□□
2. □□□□SMPP → □□□□
3. □□□□□□□□□□1-2□□□
4. □□□□□□□□□

□□□□□

□□□□□□□□□□

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

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

□□□□□□□

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

# 重启
sudo systemctl restart omnimessage-smpp
```

检查

检查

检查

1. 检查状态

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

2. 启动服务

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

3. 查看日志

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

4. 检查资源

- 检查内存
- 检查SSL证书
- 检查磁盘空间 `df -h`
- 检查系统负载 `free -h`

5. 检查网络

□□□□□□□□□□

□□□

1. □□□□□□

- □□□□SMPP → □□□□
- □□□□□□□□
- □□“□□□□”

2. □□□□□

- □□□□□□□
- □□□□□□□
- □□□□□□

3. □□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□

4. □□□□□□□

- □□□□□□
- □□□□
- □□□□□□□□□□

□□□□□

□□□□□□□□□□

□□□

1. □□**TPS**□□□

- □□□□SMPP → □□□□
- □□□□□□□□□□
- □□□□□□□□TPS□□

2. 数据库连接池

- 数据库连接池
- 数据库连接池

3. 数据库优化

- 数据库优化
- 数据库优化

4. 数据库备份

- 数据库备份
- 数据库备份

数据库

数据库

- 数据库SMPP数据库
- 数据库数据库
- 数据库数据库
- 数据库Prometheus/Grafana
- 数据库 > 98%

数据库

- 数据库
- 数据库
- 数据库
- 数据库
- 数据库

□□□□

- -
 -
 - TPS□□
 - ◆◆◆□□□□
-

□□□□

- **CONFIGURATION.md** - □□□□□□□
 - **SOURCE_ADDRESS_WHITELIST.md** - □□□□□□□□□□□□□□
 - **MONITORING.md** - □□Prometheus□□
 - **TROUBLESHOOTING.md** - □□□□□□
 - **README.md** - □□□□
-

