# Referência da API HTTP

A API OmniRoam serve arquivos TAP, arquivos RAP, visualizações de CDR, totais, controle de exportação e métricas. Este guia lista os endpoints.

## Índice


- Acesso
- Arquivos TAP
- Validação TAP
- Arquivos RAP
- Visualizações de CDR
- Upload de CDR
- Totais e Monitoramento
- Exportação TAP
- Métricas


## Acesso

A API OmniRoam escuta na porta 4000. A API serve HTTP simples, sem TLS. Todos os endpoints estão sob o caminho `/api`. A API retorna JSON sem um envelope.

| Recurso | Caminho |
| --- | --- |
| Interface do usuário Swagger | `GET /api/docs` |
| Esquema OpenAPI | `GET /api/schema` |

# Arquivos TAP

| Método e caminho | Propósito | Parâmetros |
|---|---|---|
| `GET /api/tap_files_in` | Listar os arquivos TAP de entrada. | `tadig` (opcional) |
| `GET /api/tap_files_in/:filename` | Retornar um arquivo TAP de entrada como JSON. | `tadig`, `page`, `per_page` |
| `GET /api/tap_files_out` | Listar os arquivos TAP de saída. | `tadig` (opcional) |
| `GET /api/tap_files_out/:filename` | Retornar um arquivo TAP de saída como JSON. | `tadig`, `page`, `per_page` |

A resposta do arquivo contém o cabeçalho e uma página de eventos de chamada. O valor de `page` começa em `1`. O valor de `per_page` tem um padrão de `100` e um máximo de `2000`. A resposta retorna `404` quando o arquivo não é encontrado.

# Validação TAP

| Método e caminho | Propósito | Parâmetros |
|---|---|---|
| `GET /api/tap_validate_in/:filename` | Validar um arquivo TAP de entrada. | `tadig` (opcional) |
| `GET /api/tap_validate_out/:filename` | Validar um arquivo TAP de saída. | `tadig` (opcional) |

A resposta informa se o arquivo é válido, a contagem de descobertas por severidade e a lista de descobertas. Veja Validação TAP.

# Arquivos RAP

| Método e caminho | Propósito | Parâmetros ou corpo |
|---|---|---|
| `GET /api/rap_files` | Listar os arquivos RAP. | `tadig` (opcional) |
| `GET /api/rap_files/:filename` | Retornar um arquivo RAP como JSON enriquecido. | `tadig` (opcional) |
| `POST /api/rap_files/send_alert` | Enviar um alerta por e-mail para um arquivo RAP. | Corpo: `filename` (obrigatório), `tadig` |

Veja Processamento RAP.

# Visualizações de CDR

| Método e caminho | Propósito | Parâmetros |
|---|---|---|
| `GET /api/part_cdrs/:page` | Listar as chaves de CDR parciais em voo do Redis. | Caminho: `page` (inteiro) |
| `GET /api/cdrs_from_csv` | Listar os arquivos CDR brutos ou baixar um. | `filename`, `raw_cdr_view_max_age` |
| `GET /api/config_yaml` | Retornar `config.yaml`. A API remove o token do InfluxDB. | Nenhum |
| `GET /api/config` | Retornar a seção `config` e os contadores. | Nenhum |

# Upload de CDR

| Método e caminho | Propósito | Corpo |
|---|---|---|
| `POST /api/ui_upload` | Fazer upload de um arquivo CDR CSV para o diretório OmniCore. | Multipart `file`. O nome deve ser um nome de arquivo simples terminando em `.csv` — sem separadores de caminho. Retorna 400 (rejeitado), 409 (já existente), 413 (muito grande). |

A resposta retorna `200` em caso de sucesso, `409` para um arquivo duplicado e `500` para um arquivo inválido. Veja Processamento CDR.

# Totais e Monitoramento

| Método e caminho | Propósito | Parâmetros |
| --- | --- | --- |
| `GET /api/monitoring` | Retornar totais de monitoramento ou baixar um arquivo correspondente. | `filename` (opcional) |
| `GET /api/totals_monthly` | Listar os parceiros com totais mensais. | Nenhum |
| `GET /api/totals_monthly/:tadig` | Retornar os totais mensais para um TADIG. | `year`, `month` |
| `GET /api/totals_daily` | Retornar os totais diários em todos os TADIGs. | Nenhum |
| `GET /api/totals_daily/:tadig` | Retornar os totais diários para um TADIG. | Nenhum |

# Exportação TAP

| Método e caminho | Propósito | Corpo |
|---|---|---|
| `POST /api/tap_export` | Colocar uma exportação em fila para um parceiro. | `partner` (obrigatório), `records_per_file`, `max_files` |
| `GET /api/tap_export` | Listar os trabalhos de exportação. | Nenhum |
| `GET /api/tap_export/:job_id` | Retornar um trabalho de exportação. | Nenhum |
| `GET /api/export_schedule` | Retornar o status do agendador e o estado da fila. | Nenhum |
| `POST /api/export_schedule` | Colocar uma exportação em fila para todos os parceiros agora. | Nenhum |

Os endpoints de exportação retornam `202` quando o OmniRoam coloca o trabalho em fila. A fila de trabalhos executa uma exportação por vez. Veja o Guia de Operações.

# Métricas

| Método e caminho | Propósito |
|---|---|
| `GET /api/metrics` | Retornar as métricas do Prometheus. |

A resposta é uma exposição de texto do Prometheus. Uma coleta lê um instantâneo em cache. Veja a Referência de Métricas.

---

- Visão Geral

# Arquitetura

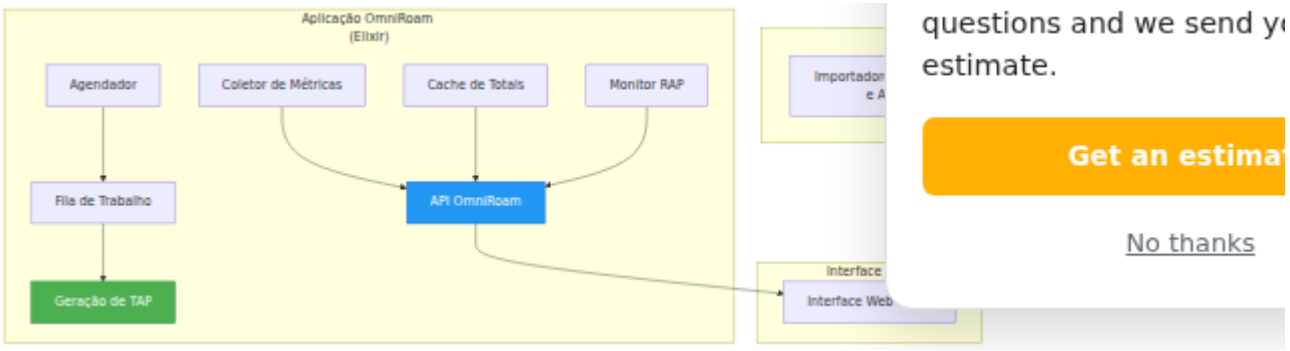
Este guia descreve os componentes do OmniRoam, o fluxo de dados de ponta a ponta e os sistemas externos com os quais o OmniRoam se integra.

# Índice


- Visão Geral do Componente
- Pacotes Implantáveis
- Fluxo de Dados
  - Ingestão e Classificação
  - Geração de Arquivo TAP
  - Transferência de Arquivo
  - Retorno RAP
- Pontos de Integração


# Visão Geral do Componente

O OmniRoam possui duas partes. Um conjunto de ferramentas Python ingere e classifica CDRs. Uma aplicação Elixir gera arquivos TAP, processa arquivos RAP e serve a API e métricas.




## Ingestão e classificação (Python)

Essas ferramentas são executadas como trabalhos agendados. Elas não são executadas como serviços de longa duração.

- **Importador OmniCore** - Lê arquivos CSV do gateway de serviço OmniCore, correlaciona os registros parciais em sessões de dados completas e grava as sessões no Redis.
- **Importador ASN.1** - Converte arquivos CDR ASN.1 3GPP para XML e os importa.
- **Importador Redis** - Drena a fila do Redis, aplica filtros de idade, arredonda o uso e envia cada registro para o Motor de Cobrança para classificação.
- **Provisionador de Tarifas** - Carrega os planos de tarifas por parceiro no Motor de Cobrança a partir de `config.yaml`.

## Aplicação OmniRoam (Elixir)

A aplicação OmniRoam é executada como um serviço systemd. Ela inicia um conjunto supervisionado de processos.

| Processo | Função |
| --- | --- |
| Índice de Arquivos | Armazena em cache os metadados dos arquivos TAP e RAP analisados para listagem rápida. |
| Cache de Totais | Armazena em cache os totais mensais e diários por parceiro. |
| Coletor de Métricas | Constrói o instantâneo de métricas do Prometheus em segundo plano. |
| Fila de Trabalho | Executa uma exportação TAP por vez. |
| Agendador | Inicia a exportação diária no horário configurado. |
| API OmniRoam | Serve a API HTTP na porta 4000. |
| Monitor RAP | Escaneia novos arquivos RAP e envia alertas. |
| Sequência | Reserva números de sequência de arquivos TAP e grava `counters.yaml`. |
| Removedor de Espaços | Limpa espaços indesejados dos campos de CDR classificados no banco de dados. |

O **Gerador TAP** lê CDRs classificados, mapeia cada CDR para um registro TAP, monta o lote, codifica o lote como um arquivo TAP binário e marca os CDRs como processados. Veja Geração de Arquivo TAP para detalhes.

## Interface web

A interface web é uma aplicação de página única React. Ela é servida como conteúdo estático pelo nginx. A interface web chama a API OmniRoam na porta 4000.

## Registrador

O registrador é um conector de saída opcional. O registrador abre um WebSocket para um sistema de gerenciamento e encaminha os pedidos desse sistema para a API OmniRoam local. Use o registrador para acessar a API de trás do NAT sem uma porta de entrada.

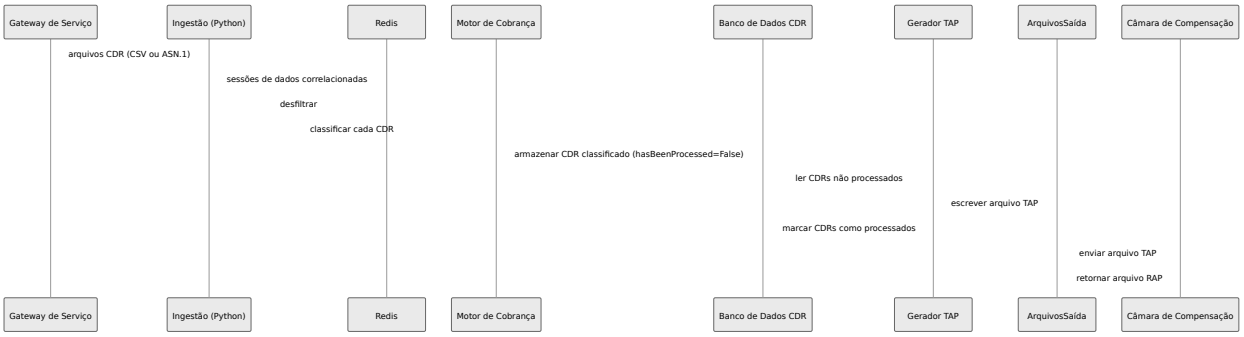
# Pacotes Implantáveis

O OmniRoam é distribuído como um conjunto de pacotes Debian. O meta-pacote `omniroam` depende dos pacotes de componente e bloqueia suas versões juntas.

| Pacote | Conteúdo | Tempo de Execução |
|---|---|---|
| `omniroam` | Meta-pacote. Depende dos pacotes de componente. | Nenhum. |
| `omniroam-api` | A versão da aplicação Elixir. | serviço systemd na porta 4000, usuário `omniroam`. |
| `omniroam-generator` | As ferramentas de ingestão e classificação Python mais a configuração. | Sem serviço. Executado pelo cron ou manualmente. |
| `omniroam-ui` | A interface web estática e o site nginx. | nginx na porta 80. |
| `omniroam-registrar` | O conector de registrador de saída. | serviço systemd. |

# Fluxo de Dados

Um CDR passa por quatro estágios: ingestão e classificação, geração de arquivo TAP, transferência de arquivo e retorno RAP.




# Ingestão e classificação

Duas fontes de CDR alimentam um pipeline.

1. **Gateways de serviço OmniCore** enviam arquivos CDR CSV. Os arquivos chegam por rsync ou por um upload HTTP para a API. O OmniRoam grava os arquivos em um diretório local.
2. **Fontes de CDR ASN.1 3GPP** fornecem arquivos CDR ASN.1. O OmniRoam puxa os arquivos e os converte para XML.
3. O importador correlaciona os registros parciais, grava sessões de dados completas no Redis e, em seguida, drena o Redis para o Motor de Cobrança para classificação.

Os CDRs classificados são armazenados no banco de dados do Motor de Cobrança (a tabela CDR). Cada CDR carrega um marcador `hasBeenProcessed`. Um valor de `False` significa que o CDR ainda precisa de um arquivo TAP. Veja Processamento de CDR para detalhes sobre a ingestão e classificação.

# Geração de arquivo TAP

O agendador ou um pedido da API coloca um trabalho de exportação na fila. A fila de trabalho executa uma exportação por vez. O gerador lê os CDRs não processados para o parceiro, do mais antigo para o mais recente, e escreve um ou mais arquivos TAP. O gerador então marca os CDRs como processados e registra o número de sequência do arquivo. Veja Geração de Arquivo TAP.

# Transferência de arquivo

Quando a fila de trabalho é drenada, o OmniRoam executa o comando de upload. O comando de upload envia os novos arquivos TAP para a câmara de

compensação. Um trabalho de pull separado recupera arquivos TAP e RAP de entrada. Veja o Guia de Operações para os trabalhos de transferência.

### Retorno RAP

O parceiro valida cada arquivo TAP. Se o parceiro rejeitar registros, o parceiro retorna um arquivo RAP. O trabalho de pull coloca o arquivo RAP no diretório de entrada. O monitor RAP escaneia o diretório, lê cada novo arquivo RAP e envia um alerta por e-mail. Veja Processamento RAP.

## Pontos de Integração

| Sistema | Protocolo | Propósito |
|---|---|---|
| Redis | Redis no localhost | Fila de CDRs parciais e correlacionados antes da classificação. |
| Motor de Cobrança | JSON-RPC na porta 2080 | Classifica CDRs e retorna o custo. |
| Banco de Dados CDR | Banco de dados na porta 3306 | Armazena CDRs classificados e os marca como processados. |
| InfluxDB | API HTTP | Armazena uso e análises de exportação. |
| Câmara de Compensação | SCP | Envia arquivos TAP e puxa arquivos TAP e RAP de entrada. |
| Provedor de E-mail | HTTPS | Envia e-mail de alerta RAP. |
| Prometheus | Coleta HTTP em `/api/metrics` | Coleta métricas do OmniRoam. |

Veja a Referência da API HTTP para os endpoints, e a Referência de Configuração para as configurações de conexão.

---


- Visão Geral

# Processamento de CDR

Este guia descreve como o OmniRoam ingere Registros de Detalhes de Chamadas (CDRs), os classifica e os armazena para a geração de arquivos TAP.

## Índice


- Fontes de CDR
- Pipeline de Ingestão
- Filtros de Idade
- Arredondamento de Uso
- Classificação
- Armazenamento


## Fontes de CDR

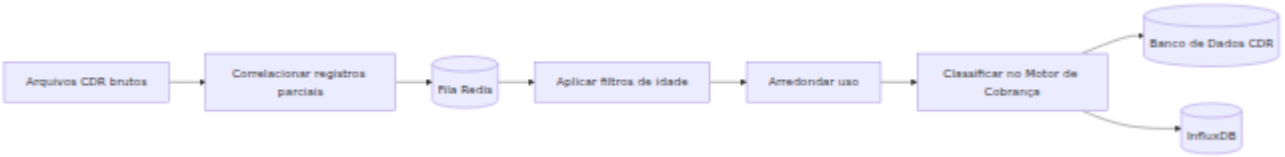
O OmniRoam ingere CDRs de dois tipos de fontes.

| Fonte | Formato | Transferência | Diretório de destino |
|---|---|---|---|
| Gateway de serviço OmniCore | CSV | rsync, ou um upload HTTP para a API | `/var/OmniCore_CDRs` |
| Fonte CDR ASN.1 3GPP | ASN.1 | rclone | `/var/asn1_cdrs` |

Um gateway de serviço também pode enviar um arquivo CSV diretamente para a API com uma solicitação `POST /api/ui_upload`. O OmniRoam salva o arquivo enviado no diretório OmniCore. Veja a Referência da API HTTP.

# Pipeline de Ingestão

As ferramentas de ingestão são executadas como trabalhos agendados. Veja o Guia de Operações para o cronograma.




1. **Correlacionar.** Uma única sessão de dados pode produzir mais de um CDR parcial. O importador junta os registros parciais de uma sessão em um registro completo.
2. **Fila.** O importador escreve cada registro completo no Redis.
3. **Filtrar.** O importador Redis esvazia a fila e remove registros que estão fora da janela de idade. O importador também remove registros que têm uso zero.
4. **Arredondar.** O importador arredonda o uso. Veja Arredondamento de Uso.
5. **Classificar.** O importador envia cada registro para o Motor de Cobrança para precificação. Veja Classificação.
6. **Armazenar.** O Motor de Cobrança armazena o CDR classificado na tabela CDR. O importador também escreve um ponto de uso no InfluxDB.

# Filtros de Idade

O importador Redis aplica dois filtros de idade antes de classificar um registro.

| Filtro | Ação |
|---|---|
| Mais velho que 30 dias | O importador remove o registro. Um arquivo TAP não pode incluir um CDR que tenha mais de 30 dias. |
| Mais jovem que 24 horas | O importador adia o registro. O importador processa o registro em uma execução posterior. |

O limite de 30 dias vem das regras da GSMA para arquivos TAP. Ingira e classifique os CDRs prontamente, ou o limite de 30 dias os excluirá da

liquidação.

# Arredondamento de Uso

O OmniRoam arredonda o uso de cada CDR para o próximo múltiplo de `round_up_to` bytes. O valor de `round_up_to` é definido por parceiro. Veja a Referência de Configuração.

O OmniRoam aplica dois arredondamentos separados.

1. **Arredondamento por direção.** O OmniRoam arredonda o volume de entrada e o volume de saída para o próximo múltiplo, cada um por conta própria.
2. **Unidades cobradas.** O OmniRoam soma os volumes de entrada e saída originais e, em seguida, arredonda esse total para o próximo múltiplo. Este total arredondado é o uso cobrado que o Motor de Cobrança precifica.

O OmniRoam mantém os volumes originais, não arredondados, para auditoria.

**Exemplo.** Um CDR tem 500 bytes de entrada e 700 bytes de saída. O parceiro usa `round_up_to: 1024`.

- Por direção: entrada arredonda para 1024, saída arredonda para 1024.
- Unidades cobradas: o total é 1200 bytes, que arredonda para 2048 bytes.

# Classificação

O OmniRoam precifica cada CDR com o Motor de Cobrança. A classificação acontece durante a ingestão, não durante a geração do arquivo TAP.

1. O provisionador de tarifas carrega a tarifa de cada parceiro no Motor de Cobrança. A tarifa vem de `unit_price` e `unit_bytes` na configuração do parceiro.
2. O importador envia cada CDR para o Motor de Cobrança com o uso cobrado.
3. O Motor de Cobrança retorna o custo. O OmniRoam armazena o custo com o CDR.

Para alterar o preço de um parceiro, atualize `unit_price` ou `unit_bytes` na Referência de Configuração e, em seguida, recarregue as tarifas no Motor de Cobrança.

# Armazenamento

O OmniRoam armazena um CDR classificado no banco de dados do Motor de Cobrança. Cada CDR possui um marcador `hasBeenProcessed`.

| Valor do marcador | Significado |
|---|---|
| `False` | O CDR está classificado, mas ainda não está em um arquivo TAP. |
| Um nome de arquivo | O CDR está no arquivo TAP nomeado. |

O gerador lê os CDRs com `hasBeenProcessed` definido como `False`, começando pelo mais antigo. Veja Geração de Arquivos TAP.

O OmniRoam também escreve análises de uso no InfluxDB. Veja a Refer��ncia de Métricas.

---

- Visão Geral

# Referência de Configuração

Este guia documenta todos os parâmetros nos arquivos de configuração do OmniRoam. O OmniRoam lê dois arquivos:

- `config.yaml` - A configuração principal. Tem uma seção global `config` e uma seção `partners` por parceiro.
- `counters.yaml` - Os contadores de sequência de arquivos TAP.

As ferramentas Python leem `config.yaml` do diretório de trabalho. O aplicativo Elixir lê `config.yaml` do caminho na variável de ambiente `OMNIROAM_CONFIG_PATH`, ou de `/etc/pytap3/config.yaml`. O OmniRoam lê `counters.yaml` do mesmo diretório que `config.yaml`.

# Índice


- Estrutura da Configuração
- Configuração Global
  - Configuração TAC
  - PLMNs de Origem
  - Identidade do Processador e Fuso Horário
  - Caminhos de Arquivos
  - Visualização de CDR Bruto
  - Conexão InfluxDB
  - Registro
  - Agendamento de Exportação
  - Comando de Conclusão de Exportação
  - Webhooks
  - Registros Por Arquivo
- Configuração do Parceiro
  - Prefixos IMSI

    - Nome do Ponto de Acesso
    - Taxas
    - Arredondamento de Uso
    - Informações de Lote
    - Informações de Contabilidade
    - Níveis de Tipo de Chamada
- Contadores de Sequência
- Chaves Obsoletas


# Estrutura da Configuração

```
config:
  # Configuração global do sistema

partners:
  # Configuração por parceiro
```

# Configuração Global

## Configuração TAC

`config.tac_config` mapeia um Código de Área de Rastreamento (TAC) para um local de atendimento. O OmniRoam usa o local de atendimento para definir o identificador de cobrança e o fuso horário em cada registro TAP. Cada entrada é um modelo nomeado com uma lista de TACs.

```
config:
  tac_config:
    ExampleNetwork:
      tac_list: ['1101', '10000', '10100']
      servingBid: 12345
      servingLocationDescription: 'Example Network'
      timezone: 'Etc/UTC'
```

| Parâmetro | Tipo | Obrigatório | Padrão | Desc |
|---|---|---|---|---|
| `tac_list` | Lista de String | Sim | - | Os valore que correspo este mod OmniRoa compara de um Cl esta lista |
| `servingBid` | Inteiro | Não | - | O identif de cobra 5 dígitos no regist Omitir pa deixar o de fora. |
| `servingLocationDescription` | String | Não | - | A descriç local esc registro T |
| `timezone` | String | Sim | - | O nome horário IA OmniRoa este fusc para calc deslocam UTC para correspo |

## PLMNs de Origem

`config.home_plmns` lista os prefixos de PLMN de origem. O OmniRoam trata um CDR cujo IMSI começa com um prefixo de PLMN de origem como um assinante

de origem, não um assinante em roaming. As ferramentas de ingestão leem esta chave.

```
config:
  home_plmns:
    - '001010'
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `home_plmns` | Lista de String | Sim | - | Os prefixos IMSI (os primeiros 6 dígitos) que identificam um assinante de origem. |

## Identidade do Processador e Fuso Horário

```
config:
  cdr_processor_tz: America/New_York
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `cdr_processor_tz` | String | Sim | - | O fuso horário IANA padrão. O OmniRoam usa este fuso horário para timestamps de arquivos e para um CDR que não corresponde a nenhum modelo TAC. |

## Caminhos de Arquivos

```yaml
config:
  cdr_search_paths:
    - '/var/asn1_cdrs/'
    - '/var/OmniCore_CDRs/'
  tap_output_path: '/etc/pytap3/OutputFiles'
  tap_human_readable_output_path: '/etc/pytap3/OutputFiles_Human'
  tap_in_path: '/etc/pytap3/TAP_In'
  roaming_in_filename: False
```

| Parâmetro | Tipo | Obrigatório | Pa |
| --- | --- | --- | --- |
| `cdr_search_paths` | Lista de String | Sim | - |
| `tap_output_path` | String | Sim | `/etc/pytap` |
| `tap_human_readable_output_path` | String | Sim | - |
| `tap_in_path` | String | Não | `/etc/pytap` |

| Parâmetro | Tipo | Obrigatório | Pa |
| --- | --- | --- | --- |
| `roaming_in_filename` | Booleano | Não | `False` |

## Visualização de CDR Bruto

```
config:
  raw_cdr_view_max_age: 7
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
| --- | --- | --- | --- | --- |
| `raw_cdr_view_max_age` | Inteiro | Não | - | A idade máxima em dias para os CDRs brutos que a interface web exibe. Um pedido pode substituir este valor com um parâmetro de consulta. |

# Conexão InfluxDB

`config.influx_db` define a conexão InfluxDB para análise de uso e exportação. Todos os quatro valores devem estar presentes, ou o OmniRoam ignora a gravação da métrica. Consulte a Referência de Métricas para as medições.

```yaml
config:
  influx_db:
    influxDbUrl: 'http://influxdb.example.com:8086'
    influxDbOrg: 'example-org'
    influxDbBucket: 'tap3'
    influxDbToken: '<influx-token>'
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `influxDbUrl` | String | Sim | - | A URL base do InfluxDB. |
| `influxDbOrg` | String | Sim | - | A organização do InfluxDB. |
| `influxDbBucket` | String | Sim | - | O bucket do InfluxDB. |
| `influxDbToken` | String | Sim | - | O token da API do InfluxDB. A API remove este valor da resposta `config_yaml`. |

Defina `influxDbToken` para seu próprio token. Não comite um token real no controle de versão.

## Registro

```yaml
config:
  log_level: DEBUG
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `log_level` | String | Sim | - | O nível de registro para as ferramentas Python. Use um dos `DEBUG`, `INFO`, `WARNING` ou `ERROR`. |

## Agendamento de Exportação

`config.export_schedule` controla a exportação automática diária. O agendador lê esta seção a cada tick, portanto, uma alteração se aplica sem reinicialização. Consulte o Guia de Operações para o comportamento do agendador.

```yaml
config:
  export_schedule:
    enabled: false
    at: '02:00'
    partners: all
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
| --- | --- | --- | --- | --- |
| `enabled` | Booleano | Não | `false` | O interruptor mestre para a exportação diária. Defina como `true` para executar a exportação automaticamente. |
| `at` | String | Não | - | O horário de execução diária no formato `HH:MM`, em UTC. |
| `partners` | String ou Lista | Não | `all` | Os parceiros a serem exportados. Use `all` para todos os parceiros, ou uma lista de nomes de parceiros. |

## Comando de Conclusão de Exportação

`config.on_export_complete` define um comando que o OmniRoam executa quando a fila de trabalhos de exportação é esvaziada. Use este comando para enviar os novos arquivos TAP para a câmara de compensação.

```yaml
config:
  on_export_complete: '/etc/pytap3/smart_tap_upload.sh'

  # Preferido: invocado uma vez por arquivo TAP pendente, com o
caminho como argumento
  # e TAP_FILE / TAP_FILENAME / TAP_TADIG / TAP_SENT_DIR no
ambiente.
  # Saída 0 e o OmniRoam move o arquivo para OutputFiles_Sent;
qualquer outro status
  # deixa pendente para tentar na próxima execução. O script não
faz glob, move arquivos,
  # ou rastreia falhas parciais — então geralmente tem dez linhas.
Tem precedência sobre
  # on_export_complete, que permanece para implantações que não
foram migradas.
  on_tap_file: '/etc/pytap3/send_one.sh'

  # Entrada: execute o script de busca do site em um cronograma em
vez de a partir do cron.
  # Invocado com TAP_IN_DIR e RAP_IN_DIR no ambiente; saída 0
significa que a
  # busca foi concluída (independentemente de ter encontrado algo
ou não). As execuções são
  # serializadas em seu próprio bloqueio, falhas são contadas, e
tap3_inbound_newest_age_seconds
  # faz uma câmara de compensação que parou de responder um alerta
em vez de uma semana silenciosa.
  # Desmarcado e nada é executado.
  on_fetch_files: '/etc/pytap3/pull_tap.sh'
  fetch_interval_minutes: 60
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `on_export_complete` | String | Não | - | O comando shell a ser executado após a conclusão da exportação. Deixe em branco para desativar. |

## Webhooks

`config.webhooks` define URLs de webhook de entrada do Slack. O OmniRoam publica uma mensagem quando uma exportação é concluída ou falha.

```
config:
  webhooks:
    on_complete: '<slack-webhook-url>'
    on_failure: '<slack-webhook-url>'
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `on_complete` | String | Não | - | A URL do webhook do Slack para uma exportação bem-sucedida. Deixe em branco para desativar. |
| `on_failure` | String | Não | - | A URL do webhook do Slack para uma exportação falhada. Deixe em branco para desativar. |

## Registros Por Arquivo

`config.records_per_tap_file` define um limite global para o número de CDRs em um arquivo TAP. Um valor por parceiro e um valor por solicitação substituem este valor global. Quando um parceiro tem mais CDRs não processados do que o limite, o OmniRoam escreve mais de um arquivo TAP.

```
config:
  records_per_tap_file: 140000
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `records_per_tap_file` | Inteiro | Não | - | O número máximo de CDRs por arquivo TAP. |

# Configuração do Parceiro

`partners` contém uma entrada por parceiro de roaming. O nome da entrada é um rótulo que você escolhe. As ferramentas de ingestão usam `imsi_prefixes` para atribuir um CDR a um parceiro. O gerador usa o nome do parceiro para selecionar o parceiro a ser exportado.

```yaml
partners:
  ExamplePartner_Live:
    imsi_prefixes:
      - '001020'
    accessPointNameOI: mnc020.mcc001.gprs
    rates:
      unit_price: 0.000476800
      unit_bytes: 1024
    round_up_to: 1024
    batch_info:
      sender: AAAAA
      recipient: BBBBB
      specificationVersionNumber: 3
      releaseVersionNumber: 12
      accountingInfo:
        localCurrency: 'USD'
        tapCurrency: 'USD'
        roundingAction: 'Simple'
        tapDecimalPlaces: 5
    call_type_level:
      qci_1: 21
      default: 20
```

## Prefixos IMSI

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `imsi_prefixes` | Lista de String | Sim | - | Os prefixos IMSI que mapeiam um CDR para este parceiro. As ferramentas de ingestão usam a correspondência mais longa. |

## Nome do Ponto de Acesso

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `accessPointNameOI` | String | Sim | - | O Identificador do Operador do Nome do Ponto de Acesso. O OmniRoam escreve este valor em cada registro TAP GPRS. |

## Taxas

`rates` define o preço que o OmniRoam carrega no Motor de Cobrança para este parceiro. O Motor de Cobrança precifica cada CDR com base nesta taxa. Consulte Processamento de CDR para o fluxo de precificação.

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `unit_price` | Float | Sim | - | O preço por cada `unit_bytes` de uso, na moeda local. |
| `unit_bytes` | Inteiro | Sim | - | O número de bytes em uma unidade de taxa. |

## Arredondamento de Uso

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `round_up_to` | Inteiro | Sim | - | A unidade de arredondamento em bytes. O OmniRoam arredonda o uso para o próximo múltiplo deste valor. Consulte Processamento de CDR. |

## Informações de Lote

`batch_info` define a identidade de cada arquivo TAP para este parceiro.

| Parâmetro | Tipo | Obrigatório | Padrão | D |
|---|---|---|---|---|
| `sender` | String | Sim | - | Seu códi caractere escreve ( `batchCo` e no nom |
| `recipient` | String | Sim | - | O código caractere OmniRoa para o su saída, o e a chav sequênc |
| `specificationVersionNumber` | Inteiro | Sim | `3` | A versão TAP. Use |
| `releaseVersionNumber` | Inteiro | Sim | `12` | A versão do TAP. U 3.12. |
| `fileTypeIndicator` | String | Não | - | Quando OmniRoa arquivo Geração Omitir es arquivo a |

## Informações de Contabilidade

`accountingInfo` define a moeda e o formato monetário de cada arquivo TAP.

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `localCurrency` | String | Sim | - | A moeda local como um código ISO 4217. |
| `tapCurrency` | String | Sim | - | A moeda de liquidação como um código ISO 4217. |
| `roundingAction` | String | Sim | - | O modo de arredondamento monetário. Use `Up`, `Down` ou `Simple`. |
| `tapDecimalPlaces` | Inteiro | Sim | - | O número de casas decimais para os valores monetários do TAP. O OmniRoam escala os valores de cobrança por 1000, então defina este valor como `3`. Consulte Geração de Arquivo TAP. |

## Níveis de Tipo de Chamada

`call_type_level` mapeia o Identificador de Classe de QoS (QCI) de um CDR para o nível de tipo de chamada TAP. O OmniRoam procura a chave `qci_<n>` para o QCI do CDR. Se a chave estiver ausente, o OmniRoam usa `default`.

```yaml
    call_type_level:
      qci_1: 21
      qci_2: 22
      default: 20
```

| Parâmetro | Tipo | Obrigatório | Padrão | Descrição |
|---|---|---|---|---|
| `qci_1` a `qci_9` | Inteiro | Não | - | O nível de tipo de chamada TAP para cada valor de QCI. |
| `default` | Inteiro | Sim | - | O nível de tipo de chamada TAP para um QCI que não tem chave `qci_<n>`. |

# Contadores de Sequência

`counters.yaml` contém os números de sequência dos arquivos TAP. O OmniRoam indexa o arquivo pelo **TADIG do destinatário**, com um contador separado para cada tipo de arquivo. Consulte Geração de Arquivo TAP para o comportamento do contador.

```yaml
BBBBB:
  CD: 11
  TD: 50
CCCCC:
  CD: 3
  TD: 2
```

| Chave | Descrição |
| --- | --- |
| Chave de nível superior | O código TADIG do destinatário de 5 caracteres. |
| `CD` | O próximo número de sequência para um arquivo ao vivo (Dados Cobráveis). |
| `TD` | O próximo número de sequência para um arquivo de teste (Dados de Teste). |

**Notas:**

- Inicialize um novo parceiro com `CD: 1` e `TD: 1`.
- O OmniRoam incrementa o contador após escrever um arquivo.
- O número de sequência é um campo de 5 dígitos. Redefina o contador antes que ele atinja `99999`. Consulte o Guia de Operações.
- Faça backup de `counters.yaml` com `config.yaml` para que a sequência continue corretamente após uma restauração.

# Chaves Obsoletas

Não use estas chaves. O OmniRoam não as lê mais no caminho de produção.

| Chave | Status |
| --- | --- |
| `config.exporters` | Obsoleta. O aplicativo atual não lê esta chave. |
| `config.cdr_processor_name` | Obsoleta. O aplicativo atual não lê esta chave. |

---


- Visão Geral

# Referência de Métricas

OmniRoam expõe duas fontes de métricas. A API do OmniRoam expõe métricas do Prometheus para o estado do serviço. As ferramentas de ingestão e exportação escrevem análises de uso no InfluxDB.

## Índice


- Métricas do Prometheus
- Status do Serviço
- Consultas de Exemplo
- Medições do InfluxDB


## Métricas do Prometheus

A API do OmniRoam serve métricas do Prometheus em `GET /api/metrics`. Uma coleta lê um instantâneo em cache, portanto, uma coleta não executa nenhuma consulta ao banco de dados. O coletor de métricas atualiza o instantâneo em segundo plano, cerca de uma vez por minuto.

Todas as métricas do OmniRoam são gauges.

| Métrica | Rótulos | Descrição |
| --- | --- | --- |
| `tap3_exporter_up` | - | O coletor tem um instantâneo atualizado. `1` significa sim. |
| `tap3_service_up` | `service` | O serviço está em execução. `1` significa ativo, `0` significa inativo. |
| `tap3_cgrates_up` | - | A API do Charging Engine está respondendo. |
| `tap3_cgrates_goroutines` | - | O número de rotinas de trabalho do Charging Engine. |
| `tap3_cdr_total` | - | O número total de CDRs no banco de dados. |
| `tap3_files_pending` | - | O número de arquivos que aguardam upload. |
| `tap3_files_sent` | - | O número de arquivos que foram enviados. |
| `tap3_files_total` | - | O número total de arquivos TAP. |
| `tap3_files_generated_today` | - | O número de arquivos gerados hoje. |

| Métrica | Rótulos | Descrição |
| --- | --- | --- |
| `tap3_last_file_generated_timestamp` | - | O tempo Unix da última geração de arquivo. |
| `tap3_last_file_sent_timestamp` | - | O tempo Unix do último arquivo enviado. |
| `tap3_partner_files_pending` | `partner` | O número de arquivos pendentes por parceiro. |
| `tap3_partner_files_sent` | `partner` | O número de arquivos enviados por parceiro. |
| `tap3_upload_status` | - | O último status de upload. `1` é sucesso, `0` é falha, `-1` é desconhecido. |
| `tap3_hours_since_last_file` | - | O número de horas desde o último arquivo TAP. |

A métrica `tap3_cdr_total` lê uma contagem aproximada de linhas. O coletor não executa uma contagem completa da tabela, porque uma contagem completa na grande tabela de CDRs esgota o pool de banco de dados de exportação.

# Status do Serviço

A métrica `tap3_service_up` relata uma série para cada dependência. O coletor verifica cada serviço com o gerenciador de serviços do sistema.

| Rótulo do Serviço | Componente |
| --- | --- |
| `omniroam-api` | A API do OmniRoam. |
| `cgrates` | O Charging Engine (rótulo emitido literalmente). |
| `mysql` | O banco de dados CDR (rótulo emitido literalmente). |
| `redis-server` | A fila de ingestão. |
| `influxdb` | O banco de dados de análises. |

# Consultas de Exemplo

```
# Arquivos TAP que aguardam upload
tap3_files_pending

# Horas desde o último arquivo TAP, alerta quando alto
tap3_hours_since_last_file > 26

# Parceiros com arquivos pendentes
tap3_partner_files_pending > 0

# Uma dependência que está inativa
tap3_service_up == 0
```

# Medições do InfluxDB

As ferramentas de ingestão e exportação escrevem quatro medições no InfluxDB. Defina a conexão na Referência de Configuração.

| Medição | Quando é escrita | Tags | Camp |
|---|---|---|---|
| `raw_cdr` | Uma vez por CDR, após a classificação. | `operator`, `input_file`, `apn`, `cellId`, `imsi`, `tac`, `sGWAddress`, `pGWAddress` | `chargeableUr` `chargedUnits` |
| `omnicore_cdr_part` | Uma vez por CDR parcial durante a ingestão do OmniCore. | `accessPointNameNI`, `hpmln`, `sgw_ip`, `tac` | `dataVolumeIr` `dataVolumeOu` |
| `file_processing` | Uma vez por arquivo do OmniCore processado. | `individual_filename` | `no_imsi_coun` `sgw_cdr_coun` `pgw_cdr_coun` `charging_id_` `roaming_sgw_` `empty_record` `processing_t` |
| `tap_cdr` | Uma vez por arquivo TAP exportado. | `operator`, `filename` | `totalcharge`, `totalconsume` `cdr_count` |

A medição `raw_cdr` tem alta cardinalidade, porque os rótulos incluem o IMSI, o identificador da célula e o nome do arquivo de entrada. Gerencie a política de retenção do InfluxDB para controlar o crescimento do armazenamento. Consulte o Guia de Operações.

---

- Visão Geral

# Guia de Operações

Este guia descreve os trabalhos agendados do OmniRoam, a transferência de arquivos, a arquivação, as localizações de logs e a manutenção rotineira.

## Índice


- Visão Geral dos Trabalhos Agendados
- Ingestão de CDR
- Exportação TAP Agendada
- Transferência de Arquivos
- Arquivamento
- Manutenção do Contador de Sequência
- Manutenção de Armazenamento
- Localizações de Logs


## Visão Geral dos Trabalhos Agendados

OmniRoam opera em um ciclo diário. A tabela mostra os principais trabalhos e sua ordem.

| Trabalho | Executa | Propósito |
|---|---|---|
| Ingestão de CDR | Agendado pelo cron | Coleta, correlaciona e classifica CDRs. |
| Exportação TAP | Diariamente, no horário configurado | Gera arquivos TAP por parceiro. |
| Upload | Após cada exportação | Envia os novos arquivos TAP para a câmara de compensação. |
| Pull Inbound | Diariamente, e uma execução completa semanal | Puxa arquivos TAP e RAP inbound. |
| Monitor RAP | A cada 4 horas | Alerta os operadores sobre um novo arquivo RAP. |
| Arquivar | Mensalmente | Despeja e purga CDRs antigos do banco de dados. |

# Ingestão de CDR

O trabalho de ingestão coleta CDRs dos gateways de serviço, os correlaciona e os classifica. O trabalho executa as ferramentas Python em ordem.

1. Puxe os arquivos CDR ASN.1 3GPP e remova arquivos locais com mais de 60 dias.
2. Converta os arquivos ASN.1 3GPP para XML e importe-os.
3. Puxe os arquivos CSV do gateway de serviço OmniCore.
4. Correlacione os registros OmniCore e escreva-os no Redis.
5. Drene o Redis, classifique cada CDR e armazene o resultado.

Consulte Processamento de CDR para detalhes do pipeline e os filtros de idade.

# Exportação TAP Agendada

O agendador inicia a exportação diária. O agendador lê `config.export_schedule` a cada tick, portanto, uma alteração se aplica sem reiniciar. Consulte a Referência de Configuração.

- O agendador executa quando `enabled` é `true`, a hora UTC atual atinge `at`, e a exportação não foi executada hoje.
- O agendador coloca em fila um trabalho de exportação por parceiro alvo.
- A fila de trabalhos executa uma exportação por vez. Este limite protege o pool de banco de dados compartilhado.
- Quando a fila é drenada, o OmniRoam executa o comando em `on_export_complete`.
- O OmniRoam envia uma mensagem no Slack para o webhook configurado ao concluir ou falhar.

O agendador não executa uma exportação extra na inicialização. Se o serviço iniciar após o horário diário, a primeira exportação será executada no dia seguinte.

Para executar uma exportação agora, use a API. Consulte a Referência da API HTTP.

# Transferência de Arquivos

OmniRoam move arquivos entre o armazenamento local e a câmara de compensação.

## Upload

OmniRoam executa o comando de upload após cada exportação, através de `on_export_complete`. O comando de upload envia os novos arquivos TAP para a câmara de compensação e move cada arquivo enviado para o diretório `OutputFiles_Sent`.

O script de upload é executado como o usuário de transferência. A câmara de compensação rejeita a chave do usuário root. Execute o upload como o usuário correto, ou a transferência falhará.

## Pull Inbound

O trabalho de pull recupera arquivos inbound da câmara de compensação para o diretório TAP inbound. O trabalho de pull copia um arquivo por vez, porque a conta da câmara de compensação não permite uma transferência em massa.

| Prefixo | Tipo de arquivo |
|---|---|
| `CD` | Um arquivo TAP ao vivo inbound. |
| `TD` | Um arquivo TAP de teste inbound. |
| `RC` | Um arquivo RAP que rejeita uma submissão ao vivo. |
| `RT` | Um arquivo RAP que rejeita uma submissão de teste. |

O trabalho de pull executa um pull incremental diário e um pull completo semanal. O trabalho de pull é executado como o usuário de transferência.

# Arquivamento

O trabalho de arquivamento controla o crescimento do banco de dados de CDR. O trabalho de arquivamento despeja os CDRs processados que são mais antigos que a janela de retenção para um arquivo e, em seguida, exclui essas linhas.

Agende o trabalho de arquivamento mensalmente. Confirme que o arquivo de despejo está completo antes de confiar na exclusão. Mantenha a janela de retenção consistente com sua política de purga de banco de dados.

# Manutenção do Contador de Sequência

O número de sequência do arquivo TAP é um campo de 5 dígitos, com um intervalo de `00001` a `99999`. O OmniRoam não reinicia o contador automaticamente.

1. Monitore os contadores `CD` e `TD` para cada parceiro em `counters.yaml`.
2. Antes que um contador atinja `99999`, concorde um reinício com o parceiro.
3. Defina o contador de volta para `1` no horário acordado.
4. Faça backup de `counters.yaml` com `config.yaml`.

Consulte a Referência de Configuração.

# Manutenção de Armazenamento

Monitore estes armazenamentos para crescimento.

| Armazenamento | Ação |
|---|---|
| `OutputFiles` | Confirme que o upload move cada arquivo enviado para `OutputFiles_Sent`. Uma contagem pendente crescente significa que o upload falhou. Observe `tap3_files_pending` e `tap3_hours_since_last_file`. |
| InfluxDB | Defina uma política de retenção. A medição `raw_cdr` tem alta cardinalidade. |
| Banco de dados de CDR | Execute o trabalho de arquivamento conforme agendado. |
| Arquivos de log | As ferramentas Python escrevem um novo arquivo de log com timestamp por execução em `/tmp`. Remova arquivos de log antigos. |

Consulte a Referência de Métricas para as métricas de monitoramento.

# Localizações de Logs

As ferramentas Python escrevem um arquivo de log com timestamp por execução.

| Padrão de arquivo | Fonte |
|---|---|
| `/tmp/import_CDR_Logger_Marben_*.log` | Os importadores ASN.1 e Redis. |
| `/tmp/import_CDR_Logger_OmniCore_*.log` | O importador OmniCore. |
| `/tmp/tap3_export_*.log` | A ferramenta de exportação legada. |
| `/tmp/convert_cdr_logger_*.log` | O conversor de 3GPP ASN.1 para XML. |
| `/tmp/rate_creator_Logger_*.log` | O provisionador de tarifas. |

A aplicação OmniRoam registra através do gerenciador de serviços do sistema. O script de upload escreve seu status em um arquivo de log que o coletor de métricas lê.

---

- Visão Geral

# Processamento de RAP

Este guia descreve os arquivos do Procedimento de Conta Devolvida (RAP) que os parceiros retornam e como o OmniRoam alerta os operadores sobre eles.

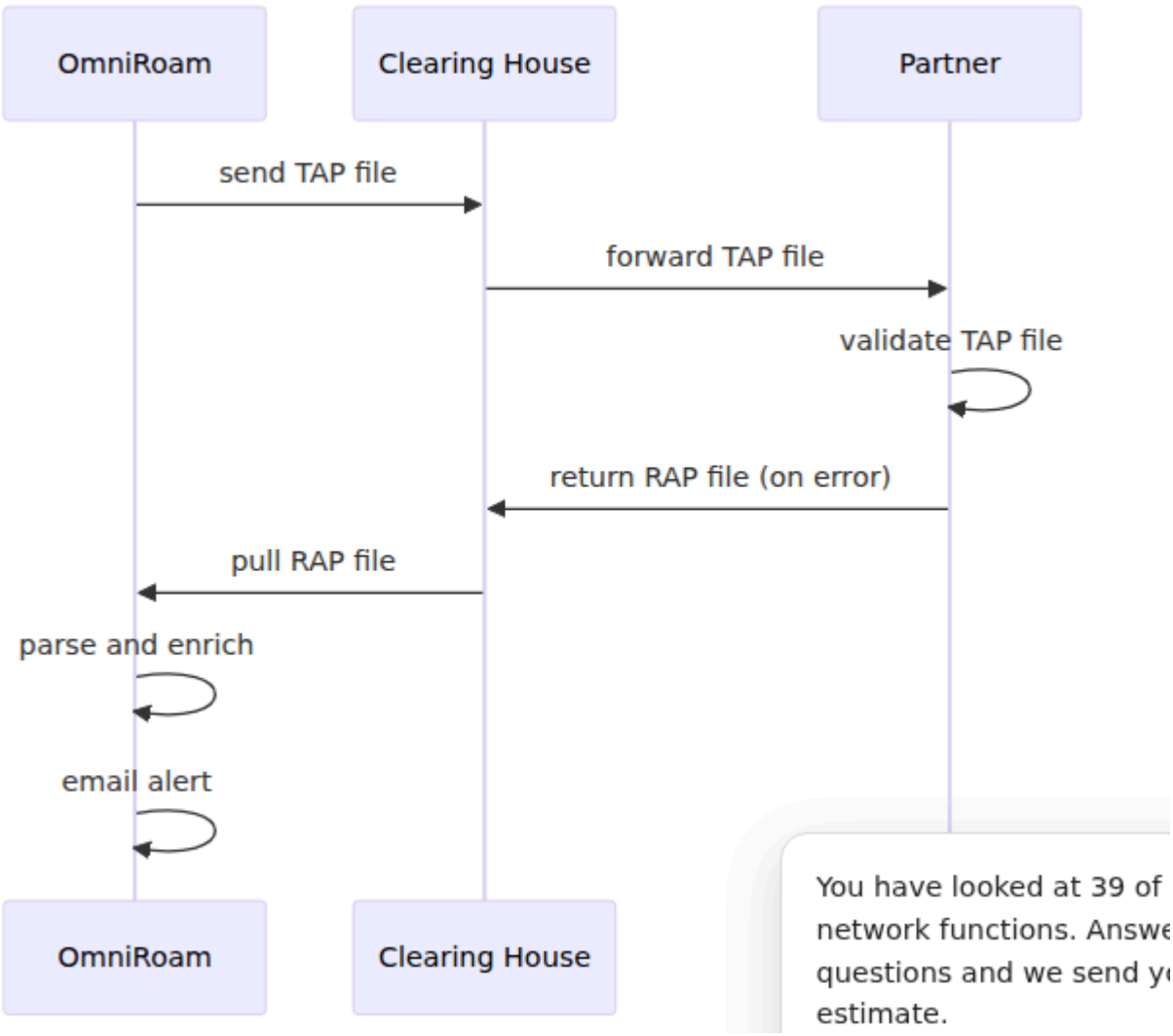
## Índice


- O que é um Arquivo RAP
- Nomeação de Arquivos RAP
- Tipos de Retorno e Severidade
- Códigos de Erro
- Monitoramento e Alertas de RAP
- Recebendo e Reconhecendo Arquivos RAP
- Reenviando um Arquivo TAP Devolvido
- Lendo um Arquivo RAP


## O que é um Arquivo RAP

Um parceiro valida cada arquivo TAP que o OmniRoam envia. Se o parceiro encontrar erros, o parceiro retorna um arquivo RAP. O arquivo RAP relata os erros em um arquivo TAP específico. O formato RAP segue GSMA TD.32.

Um arquivo RAP é o mensageiro. Os códigos de erro descrevem falhas no arquivo TAP original. O OmniRoam lê o arquivo RAP, adiciona uma descrição a cada erro e alerta os operadores.




# Nomeação de Arquivos RAP

Um nome de arquivo RAP tem um comprimento fixo de 17 caracteres. O nome espelha o nome do arquivo TAP, mas usa um prefixo RAP.

| Prefixo | Significado |
|---|---|
| `RC` | Um retorno contra um arquivo TAP ao vivo (Dados Cobráveis). |
| `RT` | Um retorno contra um arquivo TAP de teste (Dados de Teste). |

O restante do nome contém o remetente de 5 caracteres, o destinatário de 5 caracteres e o número de sequência de 5 dígitos. O OmniRoam lê apenas arquivos que começam com `RC` ou `RT` como arquivos RAP.

# Tipos de Retorno e Severidade

Um arquivo RAP contém um ou mais retornos. Cada retorno tem um tipo e uma severidade.

| Tipo de retorno | Severidade | Efeito |
|---|---|---|
| Retorno fatal | Crítico | O parceiro rejeita o arquivo TAP inteiro. |
| Retorno severo | Aviso | O parceiro rejeita os registros individuais, mas aceita o restante do arquivo. |
| Retorno ausente | Aviso | O parceiro relata uma lacuna na sequência do arquivo. |
| Retorno de parada | Informação | O parceiro relata que não recebeu nenhum arquivo TAP por 7 dias ou mais. |

Trate um retorno fatal primeiro. Um retorno fatal significa que o parceiro descartou o arquivo inteiro, e você deve corrigir a falha e reenviar.

# Códigos de Erro

O OmniRoam adiciona uma descrição a cada código de erro em um arquivo RAP. Os códigos vêm de GSMA TD.57. Os identificadores de caminho vêm de TD.57 para o arquivo TAP original e de TD.32 para o arquivo RAP.

A tabela abaixo mostra códigos de erro comuns. Veja Validação TAP para as categorias.

| Código | Descrição |
|---|---|
| 20 | Um item de dado obrigatório está ausente. |
| 50 | Um erro de codificação RAP na estrutura ASN.1. |
| 107 | Uma estrutura inválida. |
| 200 | Os dados não estão em conformidade com o acordo de roaming. |
| 250 a 257 | Um evento de chamada duplicado. |
| 260 | Um erro de idade da chamada. |
| 263 | Um formato de timestamp inválido. |

# Monitoramento e Alertas de RAP

O monitor de RAP é um processo em segundo plano na aplicação OmniRoam. O monitor observa os diretórios de entrada e alerta os operadores sobre um novo arquivo RAP.

| Propriedade | Valor |
| --- | --- |
| Intervalo de verificação | A cada 4 horas. |
| Diretórios | O diretório TAP de entrada e o diretório RAP. Um arquivo RAP chega com arquivos TAP de entrada. |
| Método de alerta | Email através do provedor de email configurado. |
| Deduplicação | O OmniRoam registra cada arquivo alertado. O OmniRoam não alerta sobre o mesmo arquivo duas vezes. |

Um alerta por email nomeia o tipo de retorno, a contagem de erros e o nome do arquivo. O alerta também inclui as informações do lote e os códigos de erro.

Defina os destinatários do alerta e as credenciais do provedor de email na configuração de tempo de execução da aplicação. Não armazene as credenciais em `config.yaml`.

Para enviar um alerta de teste para um arquivo conhecido, use `POST /api/rap_files/send_alert`. Veja a Referência da API HTTP.

# Recebendo e Reconhecendo Arquivos RAP

Quando `config.inbound_check` está definido e o papel `exporter` está habilitado, o OmniRoam recebe cada arquivo RAP e reconhecimento no diretório RAP de entrada, uma vez que não tenha mudado por `inbound_check.settle_seconds`. O diretório é verificado a cada `inbound_check.interval_seconds`.

| Arquivo | O que o OmniRoam faz |
| --- | --- |
| `RC` ou `RT` endereçado a `inbound_check.home_tadig` | Mantém o que o RAP retornou: cada retorno fatal, ausente e de parada, e os retornos severos para cada um de nossos arquivos TAP com a contagem de registros, valor e códigos de erro. Escreve o reconhecimento RAP que TD.32 requer, `AC` ou `AT` com o número de sequência do RAP, em `rap_output_path` em uma pasta para o parceiro. |
| `AC` ou `AT` para um RAP que enviamos | Marca que o RAP foi reconhecido. |

Um RAP que não pode ser reconhecido, por exemplo, porque `rap_output_path` não está definido, é tentado novamente na próxima verificação.

| Endpoint | Propósito |
| --- | --- |
| `GET /api/rap_received` | Cada arquivo RAP que um parceiro nos enviou, do mais novo para o mais antigo: seus retornos, nosso reconhecimento e qualquer arquivo TAP enviado novamente. |
| `GET /api/rap_received/:filename` | Um arquivo RAP que um parceiro nos enviou. |
| `GET /api/rap_sent` | Cada arquivo RAP que enviamos, com o reconhecimento do parceiro quando chegou. |

# Reenviando um Arquivo TAP Devolvido

Um arquivo TAP que um parceiro devolveu com um retorno fatal é enviado novamente sob o mesmo número de sequência do arquivo. TD.57 exige que ele carregue o número de sequência do arquivo RAP do RAP que o devolveu.

`POST /api/tap_resubmit` com `rap_filename`, `tap_sequence` e `commit` reconstrói o arquivo a partir da cópia que enviamos, em `tap_output_path` ou seu diretório `_Sent`. O OmniRoam adiciona o número de sequência do arquivo RAP e um novo timestamp de arquivo disponível. Sem `commit`, a resposta é uma prévia com o arquivo reconstruído. Com `commit`, o OmniRoam escreve isso em `tap_output_path` para o parceiro, onde a transferência o envia. Reenviar precisa do papel `exporter`.

Corrija a causa do retorno fatal antes de reenviar. O OmniRoam envia os mesmos registros novamente.

# Lendo um Arquivo RAP

A API do OmniRoam e a interface web leem arquivos RAP. Uma solicitação para `GET /api/rap_files/:filename` retorna o conteúdo RAP analisado e enriquecido como JSON.

O OmniRoam extrai estes campos de um arquivo RAP:

| Campo | Descrição |
| --- | --- |
| `sender` | O código TADIG da rede que enviou o arquivo RAP. |
| `recipient` | O código TADIG da rede que recebe o arquivo RAP. |
| `roamingPartner` | O código TADIG do parceiro de roaming. |
| `rapFileSequenceNumber` | O número de sequência do arquivo RAP. |
| `rapFileCreationTimeStamp` | O timestamp de criação do arquivo RAP. |
| `tapCurrency` | A moeda de liquidação do arquivo TAP original. |
| `tapDecimalPlaces` | Os lugares decimais do arquivo TAP original. |
| `returnDetailsCount` | O número de retornos no arquivo RAP. |
| `totalSevereReturnValue` | O valor total dos retornos severos. |

---


- Visão Geral

# Geração de Arquivo TAP

Este guia descreve como o OmniRoam gera arquivos TAP, como os nomeia e como gerencia os contadores de sequência.

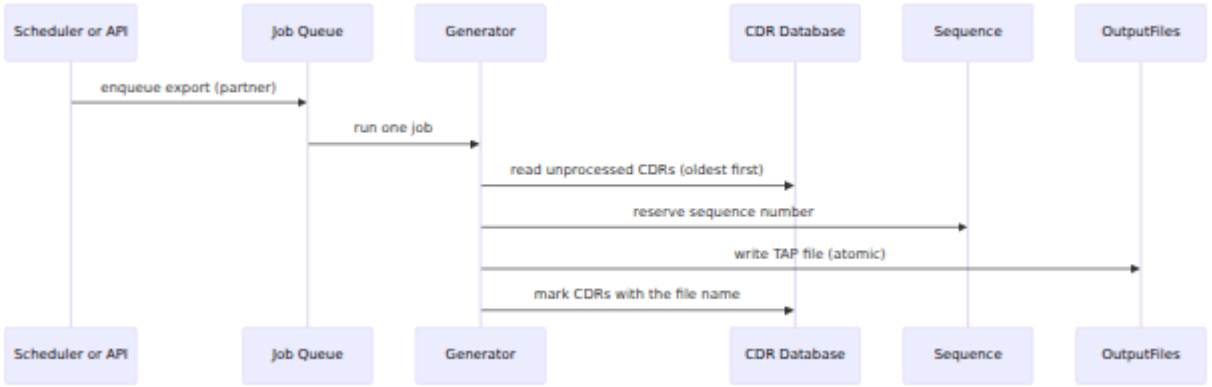
## Índice


- Visão Geral
- Como Funciona uma Execução de Geração
- Nomeação de Arquivos
- Arquivos de Teste e Arquivos ao Vivo
- Contadores de Sequência
- Estrutura do Arquivo TAP
- Valores de Cobrança
- Arquivos NOTIFY
- Acionando uma Exportação


## Visão Geral

Um arquivo TAP é um arquivo binário que relata o uso de roaming tarifado para um parceiro. O OmniRoam codifica cada arquivo TAP como ASN.1 no formato BER, de acordo com a gramática TAP 3.12. O formato segue GSMA TD.57.

# Como Funciona uma Execução de Geração




1. O agendador ou uma solicitação de API coloca um trabalho de exportação na fila para um parceiro.
2. A fila de trabalhos executa um trabalho por vez. Esse limite protege o pool de conexões do banco de dados compartilhado.
3. O gerador lê os CDRs não processados para o parceiro, do mais antigo para o mais recente.
4. O gerador divide os CDRs em arquivos se a contagem estiver acima do limite de registros por arquivo.
5. O gerador reserva um número de sequência, codifica o arquivo e o grava de forma atômica.
6. O gerador marca cada CDR no banco de dados com o nome do arquivo.

O gerador grava cada arquivo TAP em um subdiretório nomeado para o TADIG do destinatário, dentro de `tap_output_path`. O gerador também grava uma cópia JSON legível por humanos em `tap_human_readable_output_path`.

# Nomeação de Arquivos

Um nome de arquivo TAP tem um comprimento fixo de 17 caracteres.

```
CD AAAAA BBBBB 00011
```

| Parte | Comprimento | Descrição |
| --- | --- | --- |
| Prefixo | 2 | `CD` para um arquivo ao vivo, ou `TD` para um arquivo de teste. |
| Remetente | 5 | Seu código TADIG, de `batch_info.sender`. |
| Destinatário | 5 | O código TADIG do parceiro, de `batch_info.recipient`. |
| Sequência | 5 | O número de sequência com zeros à esquerda. |

O nome `CDAAAAABBBBB00011` significa um arquivo ao vivo, de `AAAAA`, para `BBBBB`, sequência `11`.

# Arquivos de Teste e Arquivos ao Vivo

A chave `fileTypeIndicator` em `batch_info` seleciona o tipo de arquivo.

- Se `batch_info` não contiver `fileTypeIndicator`, o OmniRoam produz um arquivo **ao vivo**. O nome usa o prefixo `CD` e o contador de sequência `CD`.
- Se `batch_info` contiver `fileTypeIndicator` com o valor `T`, o OmniRoam produz um arquivo **de teste**. O nome usa o prefixo `TD` e o contador de sequência `TD`.

Defina `fileTypeIndicator` como `T` enquanto você integra um novo parceiro. Remova a chave para mover o parceiro para liquidação ao vivo. Use apenas o valor `T`. Um valor diferente pode produzir um nome `TD` com um número de sequência `CD`, o que quebra o alinhamento do contador.

# Contadores de Sequência

O OmniRoam rastreia um número de sequência para cada TADIG destinatário e cada tipo de arquivo. O OmniRoam armazena os contadores em `counters.yaml`. Veja a Referência de Configuração.

- **CD** conta os arquivos ao vivo (Chargeable Data).
- **TD** conta os arquivos de teste (Test Data).

O número de sequência aumenta em um para cada arquivo. O número não aumenta para uma retransmissão. As sequências TAP e RAP são independentes.

O OmniRoam reserva um número de sequência através de um único processo e grava `counters.yaml` de forma atômica. Esse método impede que duas exportações usem o mesmo número. Uma gravação falhada mantém o arquivo de contador anterior intacto.

O número de sequência é um campo de 5 dígitos com um intervalo de `00001` a `99999`. O OmniRoam não redefine o contador automaticamente. Veja o Guia de Operações para o procedimento de redefinição.

# Estrutura do Arquivo TAP

Um arquivo TAP é uma estrutura `transferBatch`. O lote contém informações de controle e uma lista de registros de eventos de chamada. O OmniRoam produz apenas registros GPRS (dados).

| Seção | Conteúdos |
| --- | --- |
| `batchControlInfo` | O remetente, o destinatário, o número de sequência do arquivo, os timestamps do arquivo e os números da versão. |
| `accountingInfo` | A moeda local, a moeda de liquidação, as casas decimais e as informações da taxa de câmbio. |
| `networkInfo` | A tabela de deslocamento UTC e a lista de entidades de gravação (o gateway de pacotes e o gateway de serviço). |
| `callEventDetails` | A lista de registros de chamadas GPRS. |
| `auditControlInfo` | Os timestamps da primeira e última chamada, o total da cobrança e a contagem de registros. |

Cada registro de chamada GPRS contém os campos abaixo.

| Campo | Descrição |
| --- | --- |
| Assinante cobrável | O IMSI e o MSISDN quando presente. |
| Endereço PDP | O endereço IP do assinante. O OmniRoam prefere IPv6. |
| Nome do ponto de acesso | O identificador da rede e o identificador do operador do APN. |
| Timestamp de início do evento de chamada | O horário de início da cobrança, com o deslocamento UTC para a localização de serviço. |
| Duração total do evento de chamada | A duração da sessão em segundos. |
| ID de cobrança | O identificador de cobrança da sessão. |
| Informações de localização | As entidades de gravação, a área de localização, o identificador da célula e o identificador de cobrança de serviço opcional. |
| Volumes de dados | As contagens de bytes de entrada e saída. |
| Informações de cobrança | A cobrança, as unidades cobradas, as unidades tarifadas e os níveis de tipo de chamada. |

# Valores de Cobrança

Os arquivos TAP armazenam dinheiro como um inteiro. O leitor divide o inteiro por `10^tapDecimalPlaces` para obter o valor decimal.

O OmniRoam escala cada cobrança em 1000 para produzir o inteiro. Essa escala corresponde a `tapDecimalPlaces` de `3`. Defina `tapDecimalPlaces` como

`3` para cada parceiro. Um valor diferente produz um inteiro de cobrança que não corresponde às casas decimais declaradas.

A configuração `roundingAction` controla como o OmniRoam arredonda a cobrança escalada:

- `Up` arredonda o valor para cima.
- `Down` arredonda o valor para baixo.
- `Simple` arredonda o valor para o inteiro mais próximo.

# Arquivos NOTIFY

Um parceiro espera um arquivo mesmo quando não há uso. Quando um parceiro não tem CDRs não processados, o OmniRoam grava um arquivo NOTIFY. Um arquivo NOTIFY usa o mesmo formato de nome que um arquivo TAP, mas contém um registro de `notification` em vez de um lote de eventos de chamada.

# Acionando uma Exportação

Você pode iniciar uma exportação de três maneiras.

| Método | Ação |
|---|---|
| Agendador | O OmniRoam executa a exportação automaticamente no horário configurado. Veja o Guia de Operações. |
| API | Envie `POST /api/tap_export` para um parceiro, ou `POST /api/export_schedule` para todos os parceiros. Veja a Referência da API HTTP. |
| Linha de comando | Execute a ferramenta de exportação legada com o nome do parceiro. Use este método apenas como uma opção de fallback. |

---

- Visão Geral

# Solução de Problemas

Este guia lista problemas comuns e suas resoluções. Cada entrada fornece os sintomas, as possíveis causas e os passos para resolver o problema.

## Índice


- CDRs Não Chegam a um Parceiro
- Arquivos TAP Não São Carregados
- Números de Sequência Incorretos
- Um Parceiro Retorna um Arquivo RAP
- Localização Faltando em Arquivos TAP
- Valores de Cobrança Incorretos
- Sem Métricas ou Análises


## CDRs Não Chegam a um Parceiro

**Sintomas:** Um parceiro relata uso ausente. Os CDRs estão no banco de dados, mas não em um arquivo TAP.

**Possíveis causas:**

- O IMSI do CDR não corresponde a nenhum prefixo de parceiro.
- O CDR tem mais de 30 dias, então a ingestão o removeu.
- A exportação não foi executada para o parceiro.

**Resolução:**

1. Confirme que o IMSI do CDR corresponde a um prefixo em `imsi_prefixes` para o parceiro.
2. Confirme que a idade do CDR está dentro de 30 dias. Ingestione e classifique os CDRs prontamente. Veja Processamento de CDR.
3. Confirme que a exportação foi executada. Verifique os trabalhos de exportação através da Referência da API HTTP.

# Arquivos TAP Não São Carregados

**Sintomas:** A contagem de arquivos pendentes aumenta. A métrica `tap3_files_pending` está alta.

**Possíveis causas:**

- O comando de upload em `on_export_complete` está em branco ou errado.
- O upload é executado como o usuário errado, então a câmara de compensação rejeita a chave.
- A câmara de compensação não está acessível.

**Resolução:**

1. Confirme que o comando `on_export_complete` está definido. Veja a Referência de Configuração.
2. Confirme que o upload é executado como o usuário de transferência, não root.
3. Verifique o log de upload e a métrica `tap3_upload_status`.

# Números de Sequência Incorretos

**Sintomas:** Um parceiro relata uma lacuna de sequência ou um número de sequência duplicado.

**Possíveis causas:**

- O arquivo `counters.yaml` foi restaurado de um backup antigo.
- Um contador atingiu `99999`.
- O valor de `fileTypeIndicator` está definido para um valor diferente de `T`.

**Resolução:**

1. Verifique os contadores `CD` e `TD` para o TADIG destinatário em `counters.yaml`.
2. Se um contador estiver próximo de `99999`, siga o procedimento de redefinição no Guia de Operações.

3. Defina `fileTypeIndicator` como `T` para um arquivo de teste, ou remova a chave para um arquivo ao vivo. Veja Geração de Arquivos TAP.

# Um Parceiro Retorna um Arquivo RAP

**Sintomas:** O monitor RAP envia um alerta por e-mail. Um arquivo RAP está no diretório de entrada.

**Possíveis causas:**

- O arquivo TAP falhou na validação do parceiro.

**Resolução:**

1. Leia o tipo de retorno. Trate um retorno fatal primeiro, porque o parceiro rejeitou o arquivo inteiro.
2. Leia os códigos de erro no alerta ou através de `GET /api/rap_files/:filename`.
3. Corrija a falha e reenvie o arquivo. Veja Processamento de RAP.

# Localização Faltando em Arquivos TAP

**Sintomas:** Um arquivo TAP não tem identificador de cobrança de serviço ou descrição de localização.

**Possíveis causas:**

- O TAC do CDR não corresponde a nenhum modelo `tac_config`.
- O modelo não tem `servingBid`.

**Resolução:**

1. Confirme que o TAC do CDR está em uma `tac_list`. Veja a Referência de Configuração.

2. Confirme que o modelo tem um `servingBid` e uma `servingLocationDescription`.

# Valores de Cobrança Incorretos

**Sintomas:** Os valores de cobrança não correspondem ao valor de liquidação esperado.

**Possíveis causas:**

- O valor de `unit_price` ou `unit_bytes` está errado.
- O valor de `tapDecimalPlaces` não é `3`.
- O valor de `roundingAction` está errado.

**Resolução:**

1. Confirme os valores de `unit_price` e `unit_bytes`. Veja a Referência de Configuração.
2. Defina `tapDecimalPlaces` como `3`. Veja Geração de Arquivos TAP.
3. Confirme o valor de `roundingAction`.
4. Recarregue as tarifas no Charging Engine após qualquer alteração de tarifa.

# Sem Métricas ou Análises

**Sintomas:** As análises do InfluxDB estão vazias. Uma métrica do Prometheus está ausente.

**Possíveis causas:**

- A seção `influx_db` está incompleta.
- Um serviço de dependência está fora do ar.

**Resolução:**

1. Confirme que todos os quatro valores `influx_db` estão definidos. O OmniRoam ignora a gravação se algum valor estiver faltando. Veja a Referência de Configuração.

2. Verifique a métrica `tap3_service_up` para um serviço que está fora do ar. Veja a Referência de Métricas.

---

- Visão Geral

# Validação TAP

OmniRoam valida arquivos TAP contra as regras verificáveis em GSMA TD.57 e GSMA TD.32. Você pode validar um arquivo TAP de entrada ou um arquivo TAP de saída através da API.

## Índice


- Como Validar um Arquivo
- Verificações de Validação
- Severidade de Erros
- Categorias de Códigos de Erro
- Verificar um Arquivo de Entrada de um Parceiro
- Gerar um Arquivo RAP
- Monitorar um Diretório de Entrada
- Limites da Validação


## Como Validar um Arquivo

Use a API para validar um arquivo TAP.

| Endpoint | Propósito |
| --- | --- |
| `GET /api/tap_validate_in/:filename` | Validar um arquivo TAP de entrada. |
| `GET /api/tap_validate_out/:filename` | Validar um arquivo TAP de saída. |

A resposta informa se o arquivo é válido e conta as constatações por severidade. Cada constatação contém o código de erro, a severidade, a localização e uma mensagem. Veja a Referência da API HTTP.

# Verificações de Validação

OmniRoam executa quatro grupos de verificações em um arquivo TAP decodificado.

| Grupo | Verificações |
| --- | --- |
| Estrutura | As cinco seções obrigatórias estão presentes e não estão em branco: `batchControlInfo`, `accountingInfo`, `networkInfo`, `auditControlInfo` e `callEventDetails`. |
| Controle de Lote | O `sender` e o `recipient` têm 5 caracteres. O `fileSequenceNumber` tem 5 caracteres. O `specificationVersionNumber` é `3`. |
| Contabilidade | O `localCurrency` e o `tapCurrency` são códigos ISO 4217 de 3 letras. O `tapDecimalPlaces` é um número inteiro não negativo. |
| Aritmética de Auditoria | A contagem de registros é igual ao número de eventos de chamada. O total de cobrança, total de imposto e total de desconto são iguais às somas dos registros. |

# Severidade de Erros

OmniRoam relata cada constatação com uma severidade. O arquivo é válido apenas quando não há constatações fatais.

| Severidade | Efeito |
|---|---|
| Fatal | O arquivo inteiro é rejeitado. |
| Severo | O registro individual é rejeitado. O restante do arquivo é aceito. |
| Aviso | O registro é faturável. A constatação é informativa. |

# Categorias de Códigos de Erro

TD.57 agrupa os códigos de erro em intervalos numéricos.

| Intervalo | Categoria |
|---|---|
| 10 a 19 | Erro de formato ou sintaxe. |
| 20 a 29 | Valor fora do intervalo. |
| 30 a 49 | Estrutura de grupo inválida. |
| 50 a 59 | Erro geral ASN.1 ou de codificação. |
| 100 a 199 | Inconsistência entre campos. |
| 200 a 249 | Erro específico, como um duplicado, um erro IOT ou dados antigos. |
| 9000 a 9999 | Um erro bilateral, proprietário acordado entre dois parceiros. |

# Verificar um Arquivo de Entrada de um Parceiro

`POST /api/tap_inbound_check` executa cada verificação que um destinatário faz antes de faturar um arquivo TAP de um parceiro. A solicitação nomeia o arquivo, seja um já em `TAP_In` ou uma cópia carregada, e carrega o que o operador sabe sobre o relacionamento: seu próprio TADIG, os TADIGs do parceiro, seus próprios prefixos IMSI, as tarifas do acordo e a data de início do roaming comercial. Também pode carregar os prazos BA.08 e as taxas de câmbio BA.11.

A resposta fornece um veredicto para o arquivo, o status de cada verificação e cada constatação com seu código de erro TD.57 e severidade.

| Veredicto | Significado |
|---|---|
| `accept` | Sem constatações fatais ou severas. O arquivo é faturável. |
| `aged_records` | O arquivo está bom, exceto por registros além do limite de idade da chamada. Esses registros são retornados; o restante é faturável. |
| `reject_records` | Alguns registros têm constatações severas. Esses registros são retornados; o restante é faturável. |
| `reject_file` | O arquivo tem uma constatação fatal e é retornado inteiro. |
| `ignore_copy` | O arquivo é uma cópia idêntica de um já recebido. TD.57 diz para ignorá-lo. |

| Verificação | Regra | Código e severidade TD.57 |
| --- | --- | --- |
| `structure` | A estrutura, controle de lote, contabilidade e verificações de auditoria acima. | Como acima |
| `parties` | O remetente é o parceiro do acordo, e o destinatário somos nós. | 100, fatal |
| `file_name` | O tipo de nome do arquivo, remetente, destinatário e sequência correspondem ao controle de lote. | 100, fatal |
| `sequence` | O número de sequência tem 5 dígitos e não foi usado anteriormente para conteúdo diferente. Uma lacuna nomeia os arquivos ausentes. Um arquivo que retornamos com um retorno fatal pode voltar sob a mesma sequência e deve então carregar o número de sequência do arquivo RAP de nosso RAP. Um número de sequência de arquivo RAP em qualquer outro arquivo é um aviso. | 10, 20, 201 fatal; 37, 40, 200 aviso |
| `timestamps` | Os timestamps de corte e disponibilidade são válidos e estão em ordem, e o arquivo não foi disponibilizado mais de uma hora após sua chegada. | 100, 104 fatal; 101 aviso |
| `transmission` | O arquivo chegou dentro dos prazos de transmissão de arquivo BA.08. | 262 aviso; 263 fatal |

| Verificação | Regra | Código e severidade TD.57 |
|---|---|---|
| `exchange_rates` | A tabela de conversão de moeda está completa e não tem duplicatas, cada código que um registro usa está definido, e cada taxa corresponde ao BA.11. | 30 a 34, 200, 201 fatal; 101 severa |
| `call_window` | Os timestamps de chamada mais cedo e mais tarde correspondem aos registros. | 100, aviso |
| `imsi` | O IMSI de cada registro é um dos nossos. | 20, 30, 200, severa |
| `records` | O código de deslocamento UTC e a entidade gravadora de cada registro estão definidos no arquivo, e o registro não começa após o corte. | 100, severa; o corte é um aviso |
| `commercial_start` | Nenhum registro faturável está antes da data de início do roaming comercial. | 200, severa |
| `charges` | Cada cobrança corresponde à tarifa do acordo para aquele serviço e data, dentro de 0.02 unidades de moeda TAP (seção 5.8 do TD.57). As unidades cobradas correspondem ao arredondamento. | 200, 100, severa |
| `duplicates` | Nenhum registro repete um já recebido do parceiro, ou outro no mesmo arquivo. | 250, 251, 255, severa |

| Verificação | Regra | Código e severidade TD.57 |
| --- | --- | --- |
| `call_age` | Nenhum registro é mais velho que o limite de idade da chamada, 30 dias por padrão. Registros mais antigos não precisam ser aceitos, então são rejeitados, a menos que a verificação seja executada com `accept_aged`, que os mantém como registros envelhecidos. | 261 severa; 260 aviso quando aceitos |

Uma verificação cuja referência de dados a solicitação não forneceu é relatada como `not_run`, com o motivo. Nunca conta como um passe.

Cada constatação de cobrança e unidades cobradas mostra os valores esperados e reais em dinheiro ou bytes, a diferença e como o valor esperado foi calculado. Quando as unidades cobradas do parceiro são todos múltiplos inteiros de um passo maior do que o arredondamento do acordo, a constatação nomeia esse passo.

Quando o acordo tem tarifas, o bloco `money` do relatório totaliza o arquivo em unidades de moeda TAP:

| Campo | Significado |
| --- | --- |
| `declared` | O total que o parceiro cobrou. |
| `at_agreement` | Os mesmos registros precificados nas tarifas do acordo. |
| `payable` | A cobrança sobre os registros que não são rejeitados. |
| `rejected` | A cobrança sobre os registros rejeitados. |
| `rejected_at_agreement` | O custo dos registros rejeitados nas tarifas do acordo. |
| `overcharge` | `declared` menos `at_agreement`. |

# Gerar um Arquivo RAP

`POST /api/rap_generate` recebe a mesma solicitação, executa as mesmas verificações e constrói o arquivo TD.32 RAP 1.5 que retorna o arquivo TAP do parceiro.

| Constatação | Conteúdo RAP |
| --- | --- |
| Uma constatação fatal | Um retorno fatal para o arquivo inteiro. |
| Constatações severas | Um retorno severo para cada registro rejeitado, com o registro original e o contexto do erro de cada constatação. |
| Uma lacuna de sequência | Um arquivo RAP separado com um retorno ausente para a lacuna. |

Sem `commit`, a solicitação é uma prévia. OmniRoam não reserva um número de sequência RAP e não escreve um arquivo. Com `commit`, OmniRoam reserva o próximo número de sequência RAP para o parceiro e escreve o arquivo em `rap_output_path`. Dados faturáveis são retornados em um arquivo `RC` e dados de teste em um arquivo `RT`. OmniRoam não envia o arquivo RAP para o parceiro.

OmniRoam mantém cada arquivo RAP que escreve. Um arquivo TAP, ou um intervalo ausente, que já tem um RAP recebe esse RAP de volta, nunca um novo número de sequência: TD.32 diz que uma retransmissão mantém seu número. Comitar novamente escreve o mesmo arquivo de volta para `rap_output_path` se não estiver mais lá, então é enviado novamente. Cada arquivo de resposta tem `already_sent` e, para um já escrito, `written_at`.

`counters.yaml` mantém um número de sequência RAP para cada parceiro, então os arquivos RAP são enviados apenas como `config.inbound_check.home_tadig`. Uma solicitação para enviar um como qualquer outra rede é recusada.

| Configuração | Localização | Propósito |
|---|---|---|
| `config.rap_output_path` | `config.yaml` | O diretório onde os arquivos RAP comitados são escritos, em uma pasta para cada parceiro. |
| `<partner TADIG>: RC` e `RT` | `counters.yaml` | O próximo número de sequência RAP para o parceiro. Comece cada um em `1`. |

A geração de RAP precisa do papel `exporter`.

# Monitorar um Diretório de Entrada

OmniRoam pode monitorar o diretório onde os arquivos TAP dos parceiros chegam, por exemplo, um drop FTP, e verificar cada novo arquivo uma vez. Um

arquivo é verificado quando seu tamanho e hora de modificação não mudaram por `settle_seconds`, então uma transferência em andamento é deixada de lado. OmniRoam mantém cada relatório e só verifica um arquivo novamente se ele mudar.

| Configuração | Propósito |
|---|---|
| `config.inbound_check.watch_path` | O diretório a ser monitorado. |
| `config.inbound_check.home_tadig` | Nosso TADIG, que cada arquivo deve nomear. |
| `config.inbound_check.interval_seconds` | Com que frequência o diretório verificado. |
| `config.inbound_check.settle_seconds` | Quanto tempo um arquivo deve permanecer inalterado antes de ser verificado. |
| `config.inbound_check.max_call_age_days` | Registros mais velhos do que es número de dias são rejeitados, menos que aceitos como regist envelhecidos. Normalmente `30` |
| `config.inbound_check.ba08` | Opcional. Os prazos de transmissão de arquivo BA.08, `standard_transmission_hours` `exceptional_transmission_hou` |

Nossos prefixos IMSI são `config.home_plmns`. Os TADIGs e tarifas de cada remetente vêm dos acordos de gerenciamento de roaming que a sincronização de roaming salva. Um arquivo de um remetente sem acordo falha na verificação de partes, e suas cobranças não são verificadas.

| Endpoint | Propósito |
| --- | --- |
| `GET /api/tap_inbound_checks` | Os arquivos verificados, do mais novo para o mais antigo, com o remetente, veredicto e contagens. Filtrar com `sender`. |
| `GET /api/tap_inbound_checks/:filename` | O relatório completo para um arquivo. |
| `GET /api/tap_inbound_rap/:filename` | Verifica o arquivo novamente e pré-visualiza seu RAP. Nada é reservado ou escrito. Adicione `accept_aged=true` para manter registros além do limite de idade da chamada. |
| `PUT /api/tap_inbound_rap/:filename` | Verifica o arquivo novamente, reserva a sequência RAP e escreve o RAP em `rap_output_path`. Precisa do papel `exporter`. |

Ambos os endpoints RAP retornam o próprio arquivo RAP também, para que possa ser baixado e enviado ao parceiro.

| Endpoint | Propósito |
| --- | --- |
| `GET /api/tap_inbound_stop_return/:partner` | Pré-visualiza um retorno de parada para um parceiro cujos arquivos TAP pararam. Seu último número de sequência é o último arquivo recebido no caminho de monitoramento, ou `99999` quando nenhum foi recebido. Adicione `test=true` para os dados de teste do parceiro. |
| `PUT /api/tap_inbound_stop_return/:partner` | Escreve o retorno de parada. Um retorno de parada para a mesma última sequência não é numerado novamente. Precisa do papel `exporter`. |
| `GET /api/rap_sent` | Cada arquivo RAP que OmniRoam escreveu, do mais novo para o mais antigo: o parceiro, sequência, tipo de retorno e o arquivo TAP que retornou. |
| `GET /api/rap_sent/:filename` | Um arquivo RAP que OmniRoam escreveu, com o próprio arquivo em `file_b64`. |

As métricas são contadas a partir dos relatórios armazenados, então sobrevivem a uma reinicialização.

| Métrica | Rótulos |
| --- | --- |
| `tap3_inbound_check_files_total` | `sender`, `verdict` |
| `tap3_inbound_check_findings_total` | `sender`, `check`, `severity`, `code` |
| `tap3_inbound_check_records_total` | `sender` |
| `tap3_inbound_check_records_rejected_total` | `sender` |
| `tap3_inbound_check_last_timestamp` | nenhum |

Regras de alerta para arquivos e registros rejeitados, discrepâncias de cobrança, IMSIs que não são nossos, duplicatas e lacunas de sequência estão em `packaging/prometheus/omniroam-tap-inbound-alerts.yml`.

# Limites da Validação

Os endpoints de validação verificam as regras que precisam apenas do conteúdo do arquivo. A verificação de entrada adiciona as regras que precisam do acordo, dos arquivos já recebidos e dos dados de referência BA.08 e BA.11. Essas regras não são verificadas.

- As listas de códigos TADIG.
- Consultas completas de moeda ISO 4217 e números E.164.
- Tipos de chamada e faixas de tempo além dos dados de tarifas de voz e SMS do acordo.
- Quando os arquivos TAP de um parceiro pararam. Um retorno de parada é escrito manualmente com `/api/tap_inbound_stop_return`.

Use um reconhecimento de parceiro ou um arquivo RAP para confirmar o resultado da liquidação. Veja Processamento RAP.

---

- Visão Geral