# Descarga de datos vía API desde SWIFT Alliance Warehouse (v2)

## 1. Propósito y alcance

Este documento define **cómo automatizar la extracción diaria** desde **SWIFT Alliance Warehouse (AWH)** vía **API REST**, sustituyendo el proceso manual. Incluye: enlaces a máquinas, reglas de acceso, requisitos de certificados, cabeceras y endpoints, filtros a aplicar, paginación, formato de salida (**ZIP + CSV**), **nomenclatura**, **codificación/estructura exacta del CSV**, y ejemplos de prueba (curl/SOAPUI).

**Objetivo operativo**: generar por ejecución **2 ficheros** (1 .zip de mensajes brutos + 1 .csv de metadatos) y alojarlos en la ruta designada de **SUPRA Cloud**, manteniendo la compatibilidad con el proceso existente SWIFT\_parsing.

## 2. Arquitectura de conexión y acceso

### 2.1. Enlaces a máquinas y puertos

**Entorno de TEST/DEV** - Balanceador principal: warehouse\_api\_test.swift.corp — **IP** 180.18.249.15 — **Port** 48400 - Balanceador alternativo: warehouseapitest.swift.corp — **IP** 180.18.249.23 — **Port** 443

**Entorno de PRODUCCIÓN** - Balanceador principal: warehouse\_api.swift.corp — **IP** 180.18.248.27 — **Port** 48400 - Balanceador alternativo: warehouseapi.swift.corp — **IP** 180.18.248.34 — **Port** 443

**Acceso de red**: solicitar reglas de **firewall** de salida desde la IP origen de la aplicación consumidora hacia los balanceadores/puertos anteriores.

### 2.2. Autenticación y cifrado (mTLS)

* **Autenticación mutua TLS (mTLS)** mediante **certificado de aplicación cliente**.
* **Requisitos del certificado cliente**:
  + Formato: **DER**
  + Algoritmo de firma (mín.): **SHA256withRSA**
  + Suites TLS compatibles: TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384, TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
* **Flujo recomendado**:
  1. Generar keystore (ej. cliente\_api.jks) y CSR con keytool/openssl.
  2. Firmar el CSR en la **CA corporativa**.
  3. Importar el certificado **firmado** en el keystore del cliente.
  4. **Intercambio de certificados públicos**:
     + Enviar **certificado público del cliente** al equipo AWH para su alta.
     + Importar el **certificado público de AWH** en el truststore del cliente.
* **Comandos de referencia** (adaptar rutas/nombres/contraseñas):
* # Generar par de claves y keystore (JKS)  
  keytool -genkeypair -keystore cliente\_api.jks -sigalg SHA256withRSA -keyalg RSA  
    
  # Generar CSR  
  keytool -certreq -keystore cliente\_api.jks -file cliente\_api.csr  
    
  # Importar certificado firmado  
  keytool -importcert -keystore cliente\_api.jks -file cliente\_api.crt -alias cliente\_api  
    
  # Exportar certificado público para AWH  
  keytool -exportcert -keystore cliente\_api.jks -file cliente\_api.cer -alias cliente\_api  
    
  # Importar certificado público de AWH en el truststore del cliente  
  keytool -importcert -keystore cliente\_api.jks -file awh\_api.cer -alias awh  
    
  # Listar contenido del keystore  
  keytool -list -v -keystore cliente\_api.jks

**SoapUI**: en **Preferences → SSL Settings** seleccionar el keystore del cliente y su password. En cada request, verificar el certificado en **SSL Info**.

### 2.3. Cabeceras obligatorias y control de aplicación

* **Headers requeridos** en todas las llamadas:
  + Version: '<version\_de\_la\_aplicacion>'
  + Application: '<nombre\_aplicacion\_registrada\_en\_AWH>'

## 3. Endpoints, filtros y paginación

### 3.1. Endpoint base

* Esquema: https
* Host/puerto: según entorno (ver §2.1)
* Base path: típicamente /v1/messages (confirmar en el descriptor OpenAPI/YAML entregado por AWH).

### 3.2. Criterios de búsqueda a reproducir (equivalentes a la UI)

Para **cada ejecución diaria** se deben aplicar los filtros que actualmente se marcan manualmente en la interfaz: - **Creation date**: desde 00:00:00 hasta 23:59:59 del **día a descargar**. - **Message type**: incluir **todos los tipos** usados en la UI (mantener lista actualizada en configuración). - **Status**: únicamente Sent to network **o** Received. - **Direction**: resultado puede incluir **Incoming (from SWIFT)** y/o **Outgoing (to SWIFT)**.

**Nota**: la API devuelve **JSON** con un campo total\_count para el conteo de coincidencias.

### 3.3. Paginación / Scroll

* La API soporta **paginación y/o scroll**.
* Configurar **limit** y **offset** (o token de scroll) para iterar hasta consumir todos los resultados del día.
* **Buenas prácticas**:
  + Usar **ventanas deterministas** (orden por creation\_date, warehouse\_id como desempate) para evitar duplicidades.
  + Implementar **reintentos idempotentes** y **resumen de control** (totales esperados vs. descargados).

## 4. Formato de entrega y nomenclatura

Cada ejecución debe producir **dos ficheros** con la **misma execution\_date** (YYYYMMDD):

1. **ZIP de mensajes brutos**
   * **Nombre**: report\_{execution\_date}.zip
   * **Contenido**: mensajes completos (raw views) correspondientes a los filtros del día.
2. **CSV de metadatos**
   * **Nombre**: dates\_{execution\_date}.csv
   * **Columnas**: exactamente Warehouse ID,Direction,Creation date (en ese orden)

**Destino**: ambos ficheros deben alojarse en la ruta acordada de **SUPRA Cloud** para su posterior procesamiento por SWIFT\_parsing.

## 5. Especificación **exhaustiva** del CSV (debe cumplir exactamente)

A partir del CSV de ejemplo validado (enero 2025):

* **Tipo de archivo**: **texto plano** (text/plain)
* **Delimitador**: **coma** (,)
* **Citas/quoting**: " como **quotechar** estándar, sin escapechar explícito (no se esperan comillas en campos actuales)
* **Separador decimal**: . (si procede)
* **Cabecera**: **sí** (primera línea con nombres exactos de columnas)
  + Warehouse ID,Direction,Creation date
* **Codificación**: **UTF‑8** (ASCII-safe)
* **Saltos de línea**: **CRLF** (\r\n)
* **Envoltorio**: **sin BOM**
* **Tamaño orientativo**: ~5 MB por día (puede variar con el volumen de mensajes)
* **Cardinalidad**: una fila por mensaje recuperado por la búsqueda
* **Dominio de valores**:
  + Warehouse ID: cadena alfanumérica (ej.: 1862AA7FFF3FA8C9OUTPUT)
  + Direction: Incoming (from SWIFT) **o** Outgoing (to SWIFT)
  + Creation date: **dd-MMM-YYYY HH:mm:ss** (mes en **inglés** abreviado; zona horaria inherente al sistema AWH, sin offset en campo)
    - Ej.: 31-Jan-2025 23:58:24

**Validación de muestra (enero 2025)**: 75.243 filas; rango de Creation date entre 2025-01-01 00:07:06 y 2025-01-31 23:58:24.

**Requisitos de exportación equivalentes en UI** (para trazabilidad): - Change Columns → incluir **sólo**: Warehouse ID, Direction, Creation Date. - Format: **CSV** - Table Columns: **Previewed columns** - **Excluir**: Application Header, Message Body, Detailed history.

## 6. Procedimiento de descarga (lógica de extracción)

1. **Construir query** aplicando filtros de §3.2 para execution\_date (ventana 00:00–23:59:59).
2. **Paginar**/scrollear hasta recuperar todos los mensajes del día, controlando total\_count.
3. **Generar**:
   * **ZIP** con *raw views* de todos los mensajes del día.
   * **CSV** con las **tres columnas** especificadas, en **UTF‑8**, delimitado por **coma**, **CRLF**.
4. **Renombrar** ambos con la nomenclatura de §4 y **subir a SUPRA Cloud** en la ruta convenida.
5. **Controles**: hash (MD5/SHA256), recuento de filas vs. total\_count, y logging de ventana temporal y parámetros.

## 7. Ejemplos de integración

### 7.1. Ejemplo curl con mTLS

# Variables de entorno  
AWH\_HOST="warehouse\_api\_test.swift.corp:48400" # o el de PROD  
APP\_NAME="<nombre\_app\_registrada>"  
APP\_VERSION="<version\_app>"  
CLIENT\_CERT="/ruta/cliente\_api.crt" # certificado público cliente  
CLIENT\_KEY="/ruta/cliente\_api.key" # clave privada cliente  
CA\_CHAIN="/ruta/awh\_api.cer" # certificado público de AWH (o cadena CA)  
  
# Filtros (ejemplo)  
START="2025-10-01T00:00:00Z"  
END="2025-10-01T23:59:59Z"  
OWN\_BIC="BSCHESM0" # ejemplo  
STATUS="Sent to network,Received"  
  
curl -sS --fail \  
 --cert "$CLIENT\_CERT" --key "$CLIENT\_KEY" --cacert "$CA\_CHAIN" \  
 -H "Application: '$APP\_NAME'" \  
 -H "Version: '$APP\_VERSION'" \  
 "https://$AWH\_HOST/v1/messages?own\_bic=$OWN\_BIC&creation\_date\_from=$START&creation\_date\_to=$END&status=$STATUS&limit=1000"

### 7.2. Configuración en SOAPUI (resumen)

1. Importar el **YAML/OpenAPI** de AWH en SOAPUI 5.7+.
2. En **Request** introducir https://<host>:<port> y el path del recurso.
3. Añadir headers Application y Version.
4. En **SSL Settings**, seleccionar el **keystore** del cliente.
5. Probar llamada y revisar **total\_count** y contenido JSON.

## 8. Consideraciones operativas y de seguridad

* **Rotación de certificados**: calendarizar aviso previo a expiración (30/15/7 días).
* **Tiempos de respuesta y reintentos**: política de backoff exponencial para 5xx/timeouts.
* **Idempotencia**: clave natural warehouse\_id + creation\_date para evitar duplicados en reintentos.
* **Trazabilidad**: registrar parámetros de búsqueda, ventana temporal, total\_count, páginas descargadas y hashes de ficheros generados.
* **Límites**: ajustar limit según recomendaciones de AWH y capacidad de red.

## 9. Checklist para el equipo de Ingestas

* Reglas de firewall a **TEST**/**PROD** (hosts/puertos §2.1)
* Keystore y CSR generados; certificado **firmado por CA** importado (§2.2)
* Certificado público **AWH** importado en truststore (§2.2)
* Headers Application/Version configurados (§2.3)
* Filtros de consulta equivalentes a UI (§3.2)
* Paginación/scroll implementados (§3.3)
* Generación de **ZIP** y **CSV** conforme a §5
* Nomenclatura report\_YYYYMMDD.zip y dates\_YYYYMMDD.csv (§4)
* Subida a ruta de **SUPRA Cloud**
* Controles de integridad y reconciliación con total\_count

## 10. Anexo – Mapeo UI ↔ API

* **Creation date (UI)** → creation\_date\_from / creation\_date\_to (API)
* **Message type (UI)** → message\_type (lista configurable)
* **Status (UI)** → status (valores: Sent to network, Received)
* **Direction (UI)** → derivado de direction/backend\_direction
* **Export CSV (UI)** → generación del CSV con columnas exactas (véase §5)

## 11. Historial de cambios

* **v1**: Documento inicial (extracción manual → API).
* **v2**: *Este documento*. Añadidos: **enlaces a máquinas/puertos**, **certificados mTLS**, **headers obligatorios**, **paginación**, y **especificación exhaustiva del CSV (delimitador, codificación, CRLF, formato de fecha, dominios de valores)**; ejemplos curl/SOAPUI; checklist de ingestas.