

¿Cómo integrar el parseo completo antes del envío?

1. Tu módulo de parsing del archivo ISO 20022 crea un List<DetalleRequest>. Ejemplo simplificado:

```
public List<DetalleRequest> parsearDetallesComoDTO(String rutaArchivoISO, String idEncabezado) {
    List<DetalleRequest> listaDTO = new ArrayList<>();
    // Recorres tu archivo ISO 20022, validas cada registro y extraes sus 25 campos:
    // Por cada registro extraído:
    DetalleRequest dr = new DetalleRequest(
        idEncabezado,
        valorDetalleId,
        valorCampo1,
        valorCampo2
        // ... el resto de valores
    );
    listaDTO.add(dr);
    // ...
    return listaDTO;
}
```

2. En tu flujo principal (MainApp u otra clase), harías:

```
public class MainApp {
    public static void main(String[] args) throws Exception {
        APIRestComponentLogger apiLogger = new APIRestComponentLogger();

        // 1. Parsear encabezado y enviar → obtienes idEncabezado
        Map<String, Object> encabezado = parsearEncabezado(archivoISO);
        GBDRspIngresarACHRecibidosRest respHdr =
            apiLogger.invokeRestServices(urlEncabezado, encabezado, useSSL);
        String idEncabezado = respHdr.getId();

        // 2. Parsear lista completa de detalles a POJOs
        List<DetalleRequest> listaDTO = parsearDetallesComoDTO(archivoISO, idEncabezado);

        // 3. Enviar detalles con multihilo/batch segun config
        apiLogger.processDetallesDTO(
            listaDTO,
            urlDetalle,
            useSSL
        );
    }
}
```

Con esto, todo el armado del objeto (POJO) ocurre en el parseo (antes de entrar en processDetallesDTO), y tu clase APIRestComponentLogger solo se encarga de serializar el POJO o la lista de POJOs y hacer la llamada HTTP+JSON. Así, es reutilizable para cualquier otro endpoint que reciba un POJO similar.