***Plan van aanpak***

***XML TO JSON Converter watchdog***



### **Inhoudsopgave**

1. Inleiding
2. Doelstelling
3. Beschrijving oplossing – Optie 1: Watchdog tool  
   3.1 Functionaliteit  
   3.2 Configuratiebestand  
   3.3 Logging  
   3.4 Verwerkingsproces
4. Beschrijving oplossing – Optie 2: Middleware  
   4.1 Functionaliteit  
   4.2 Processtappen
5. Wat is Supabase?
6. Technische Specificaties
7. Conclusie

### 1. **Inleiding**

Dit plan van aanpak beschrijft twee opties voor het automatisch converteren van XML-bestanden naar JSON-formaat, gekoppeld aan Supabase als opslag- en distributieplatform. De oplossingen zijn bedoeld ter ondersteuning van geautomatiseerde gegevensverwerking tussen AGP-ERP en externe partijen (zoals grossiers of leveranciers).

### 2. **Doelstelling**

Het doel is het realiseren van een tool die automatisch XML-bestanden converteert naar JSON-bestanden en deze opslaat in Supabase. De tool moet kunnen detecteren wanneer AGP-ERP een XML-bestand aanlevert, dit automatisch omzetten, opslaan en beschikbaar maken voor verdere verwerking of distributie.

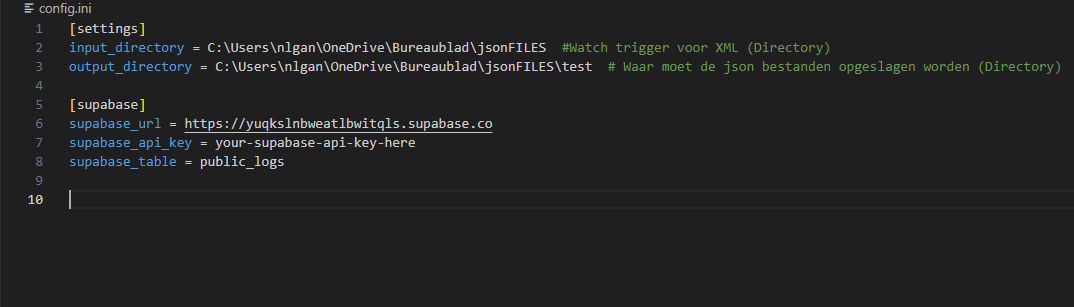
### 3. **Beschrijving oplossing – Optie 1: Watchdog tool**

#### 3.1 Functionaliteit

De tool (‘toolJson.py’) monitort een specifieke directory. Zodra er een XML-request en -response gedetecteerd worden, converteert de tool deze bestanden automatisch naar JSON en slaat deze op in een opgegeven pad of in Supabase.

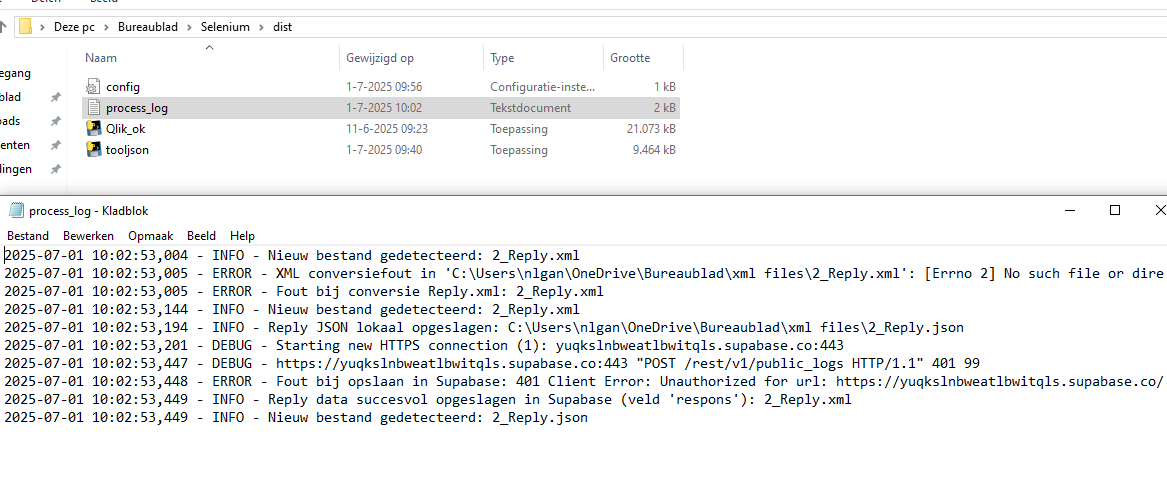
#### 3.2 Configuratiebestand

De tool vereist een configuratiebestand (config.ini) waarin het pad wordt opgegeven waar de JSON-bestanden opgeslagen moeten worden. Dit pad kan lokaal zijn of verwijzen naar een Supabase-verbinding. Bij een onjuiste configuratie start de tool niet.



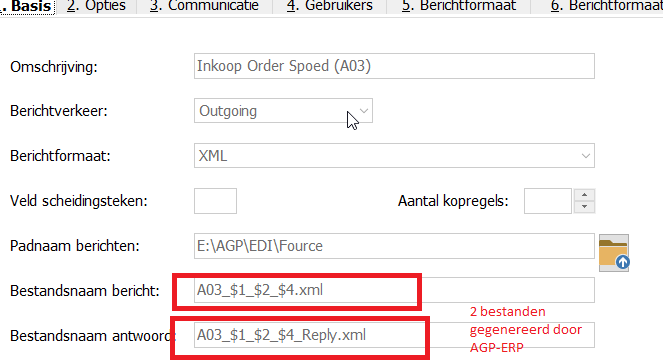
#### 3.3 Logging

De tool genereert een proceslogbestand waarin alle acties worden bijgehouden, inclusief bestandsnamen, verwerkingstijden en foutmeldingen. Dit is essentieel voor debugging en monitoring.

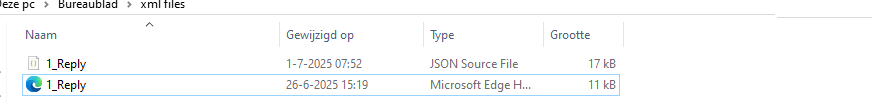


#### 3.4 Verwerkingsproces

* De watchdog detecteert XML-bestanden in de rootmap.



* Beide XML-bestanden (request & response) worden gelezen.
* Conversie naar JSON wordt uitgevoerd.



* JSON-bestanden worden opgeslagen op de opgegeven locatie.

### 4. **Beschrijving oplossing – Optie 2: Middleware**

#### 4.1 Functionaliteit

De middleware is een webgebaseerde API die inkomende XML-berichten ontvangt, deze converteert naar JSON en opslaat in Supabase. Het systeem fungeert als brug tussen AGP-ERP en andere gekoppelde systemen.

#### 4.2 Processtappen

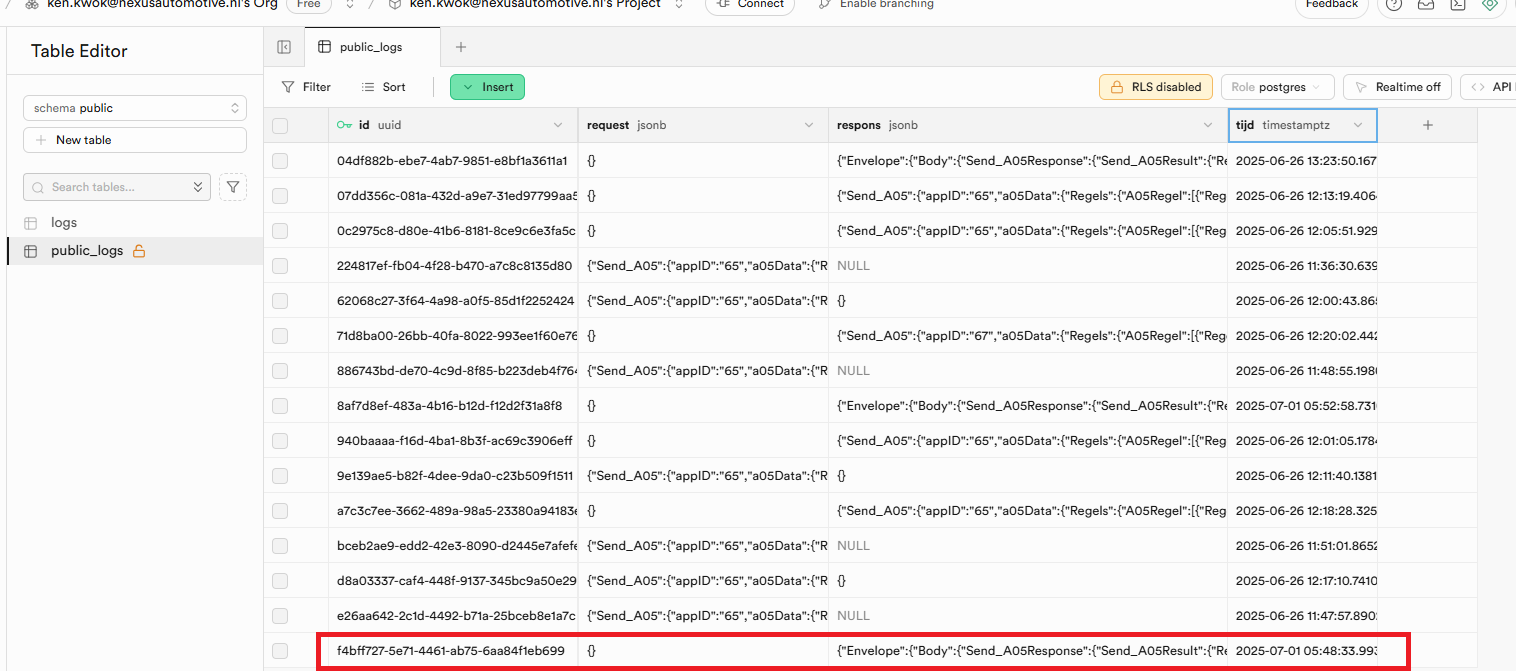
1. AGP-ERP stuurt een XML-bericht via een HTTP POST-aanroep.
2. Middleware ontvangt het bericht.
3. XML wordt automatisch geconverteerd naar JSON.
4. JSON wordt opgeslagen in Supabase:
   * requests: voor inkomende aanvragen.
   * responses: voor terugkoppelingen indien aanwezig.
5. Andere systemen kunnen de JSON-gegevens opvragen of verwerken via API-integratie.

### 5. **Wat is Supabase?**

Supabase is een open-source alternatief in de cloud, met PostgreSQL als achterliggende database. Het biedt functionaliteiten zoals:

* Databasebeheer
* Gebruikersauthenticatie
* Bestandsopslag
* RESTful API-integraties

Voor dit project fungeert Supabase als opslag- en distributieplatform voor geconverteerde JSON-data, zodat deze efficiënt gedeeld kan worden met andere systemen.



### 6. **Technische Specificaties**

* **API-methode (optie 2)**: HTTP POST

### **7. Conclusie**

Van de twee oplossingen is “Optie 1” (de Watchdog-tool) het meest geschikt en is al geimplementeerd de tool zit op de root -> Qlikdata/SOFTWARE/toolJson.exe, tenminste als deze aansluit bij de eisen en wensen van de grossier. Deze optie is al ontwikkeld en functioneert als een zelfstandige service die XML-bestanden automatisch omzet naar JSON en deze eventueel ook kan opslaan in Supabase. De tool is eenvoudig te configureren, werkt betrouwbaar en vereist geen extra infrastructuur of complexe integraties.

“Optie 2” (de middleware-oplossing) biedt op termijn meer flexibiliteit door middel van een webgebaseerde API-integratie, maar vergt aanzienlijk meer ontwikkeltijd. Deze middleware moet volledig worden opgebouwd in C# .NET API, inclusief functies voor API-authenticatie, foutafhandeling en integratie met Supabase. Hierdoor is optie 2 op dit moment minder realistisch binnen de beschikbare tijd en middelen.

Daarom heeft optie 1 onze voorkeur, omdat deze de projectdoelstellingen al volledig ondersteunt en sneller in productie kan worden genomen.