delight REST-API

Dokumentation für Entwickler

Einleitung

Delight Business Software Produkte enthalten eine REST-API für den externen Zugriff und die Integration von 3.-Herstellersoftware.

Dieses Dokument richtet sich explizit an Software-Entwickler und setzt grundsätzliche Kenntnisse in den Bereichen REST, http, SSL, JSON und XML voraus.

Der lokale REST-API Testclient ist eine Standalone-Installation. Die REST-API läuft bei dieser Variante im CRM-Client. D. h. die Schnittstelle ist nur verfügbar, solange der Client läuft. Im produktiven Umfeld läuft die gleiche API im delight Windows-Service – unabhängig von den Clients.

Technologie

Die REST-API läuft über http oder https (entweder oder, keine parallelbetrieb möglich) und setzt auf dem Kernel Modul http.sys auf. Entsprechend kann die API sehr viele Requests handeln. Damit die API im schnellen http.sys Modus läuft, muss der Client einmal als (Windows)Administrator gestartet werden. Ansonsten läuft die API im Fallback-Modus auf Sockets. Für die Entwicklung spielt das in der Regel aber keine Rolle.

Abgrenzung

Diese Dokumentation beinhaltet eine grundlegende API-Installations- und Funktionsbeschreibung. Auf eine Dokumentation über spezifische Inhalte verzichten wir bewusst. An Stelle einer fachlichen Inhaltsdokumentation stellen wir eine komplette, lauffähige API-Umgebung, mit konkreten Beispielen, die live getestet, angepasst und ausgeführt werden können zur Verfügung.

Anmerkung Bilder

Unter jedem Bild befindet sich ein Link zum jeweiligen Bild in original Grösse.

SSL

SSL funktioniert nur im schnellen http.sys Modus. Der Socket-Fallback unterstützt kein SSL.

Live Beispiel-Calls mit dem REST-Debugger

Mit dem RESTDebugger kann die API "live" getestet und damit rumgespielt werden. Im Archiv ist ein Preset-File mit vielen, konkreten Beispielen zu den wichtigsten Prozessen enthalten.

Installation lokaler Test-Client

Download **aktuelle** Version **2023**:
https://mirror01.delight.ch/crmrestapi/RESTDeveloperClient2023.zip

Download Vorgänger Version 2022:
https://mirror01.delight.ch/crmrestapi/RESTDeveloperClient2022.zip

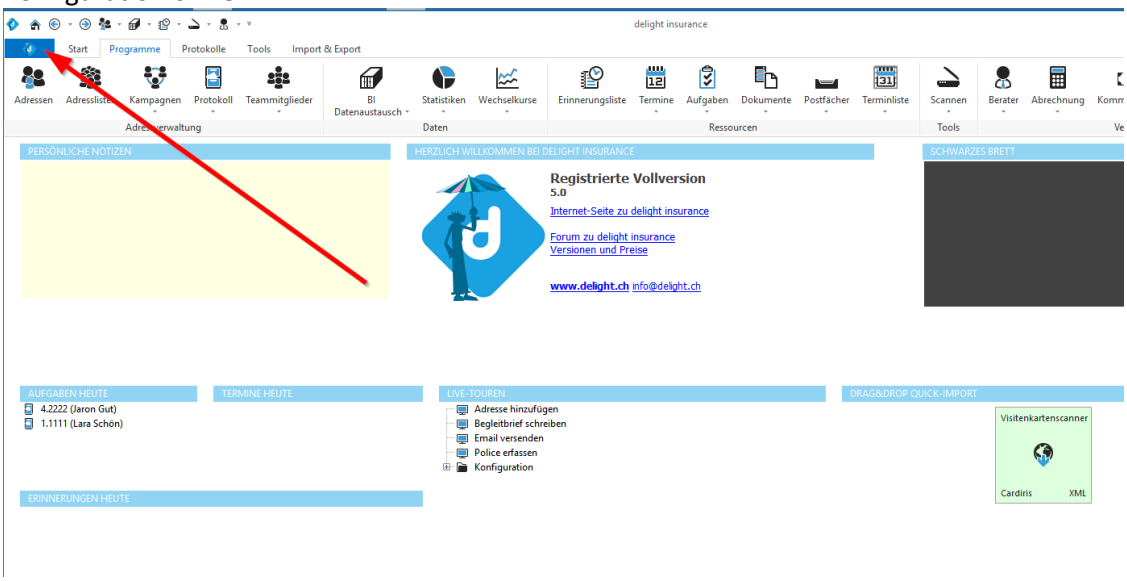
ZIP downloaden und entpacken.

delight REST-Service konfigurieren

(einmalig, wenn das erledigt ist, reicht es den Client zu starten)

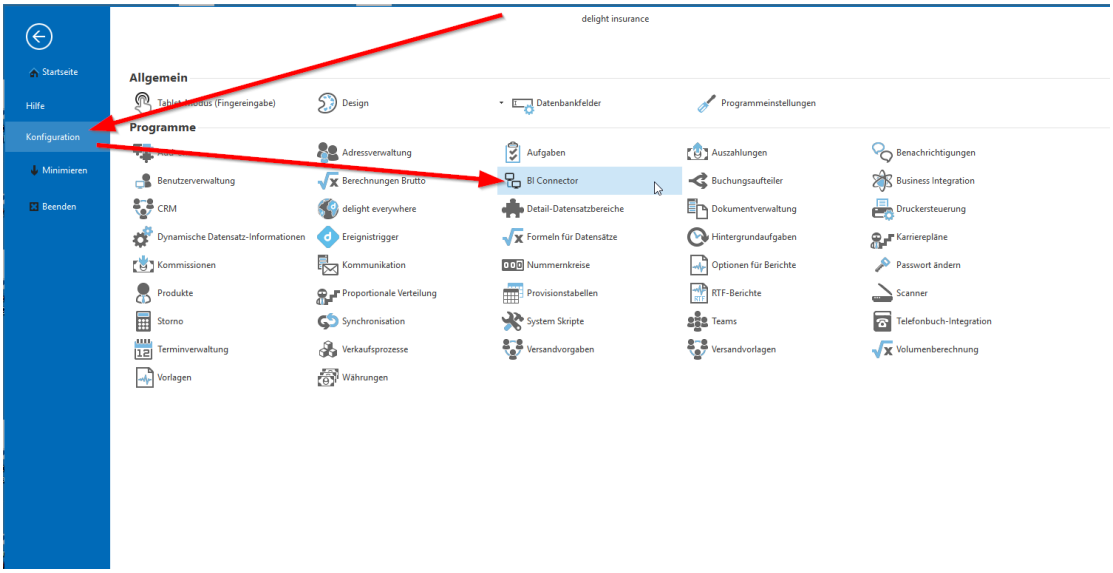
1. CRMWSTestSystem_x64\ML2Client.exe starten und anmelden mit
   Benutzername: admin
   Passwort: admin

2. Konfiguration öffnen



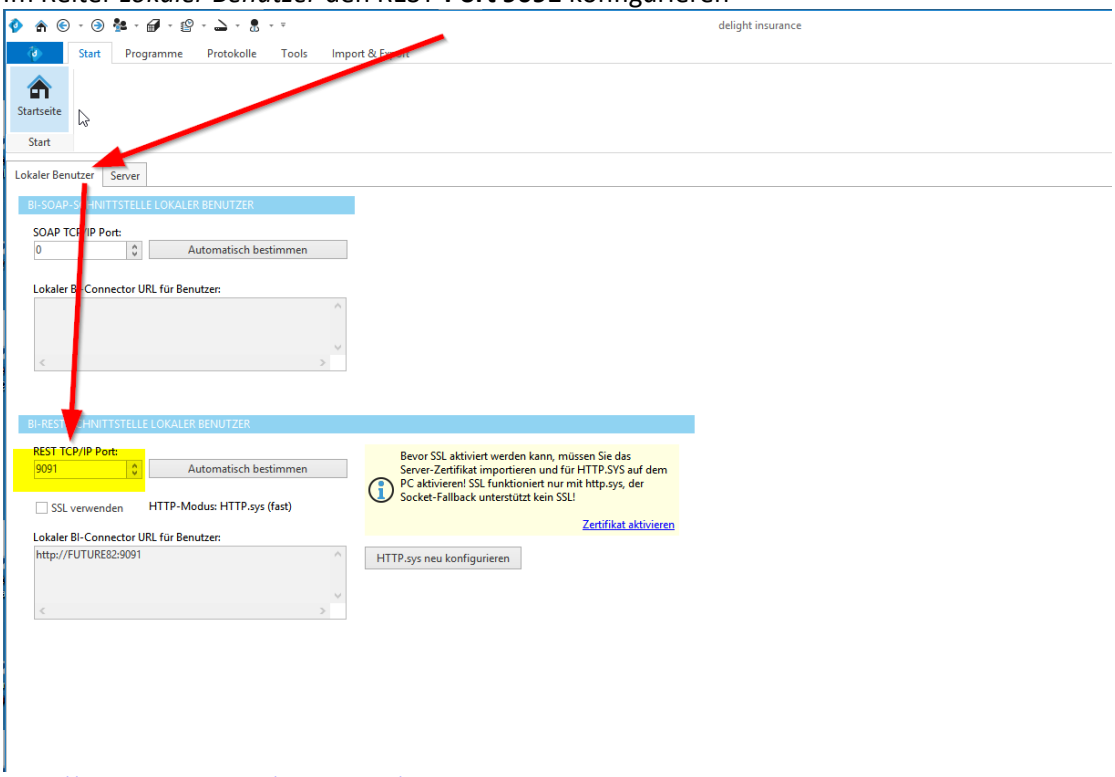
https://mirror01.delight.ch/supportfiles/delight_4000199715.png

3. Konfiguration > BI Connector öffnen



https://mirror01.delight.ch/supportfiles/delight_4000199855.png

4. Im Reiter *Lokaler Benutzer* den REST-**Port 9091** konfigurieren



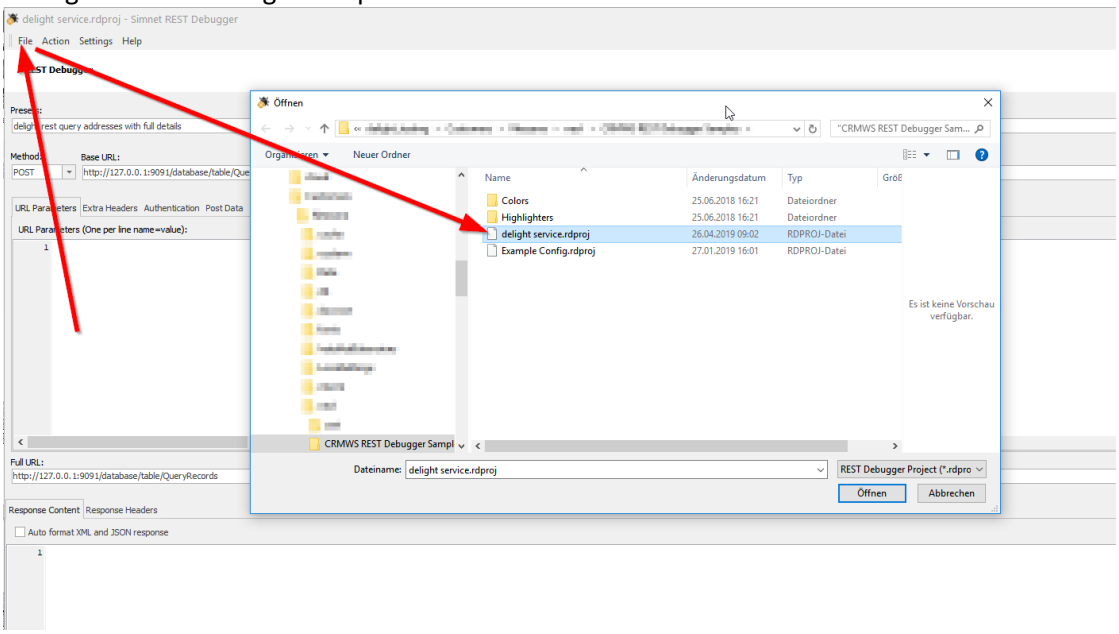
https://mirror01.delight.ch/supportfiles/delight_4100199918.png

RESTDebugger Preset laden

(einmalig, der RESTDebugger merkt sich das letzte verwendete Preset)

1. RESTDebugger.exe starten

2. Konfiguration mit delight Beispiel-Presets laden.



https://mirror01.delight.ch/supportfiles/delight_3800199642.png

Mit dem REST Debugger auf delight REST-API zugreifen

Login (SessionKey)

Die REST-API erlaubt ein paar wenige Calls ohne Authentifizierung. Grundsätzlich muss zuerst per Login-Request eine Session eröffnet werden. Bei der Benutzer-Anmeldung wird eine Session eröffnet und der SessionKey zurückgeliefert. Der SessionKey muss bei jedem folgenden API-Request im HTTP-Header mitgeliefert werden.

Nach Abschluss der Kommunikation sollte die Session grundsätzlich per Logout-Request wieder beendet werden. Der Server schliesst inaktive Sessions nach einer gewissen Zeit automatisch.

1. CRM Client starten (Datei ML2Client.exe starten) und anmelden mit:
   Benutzername: admin
   Passwort: admin

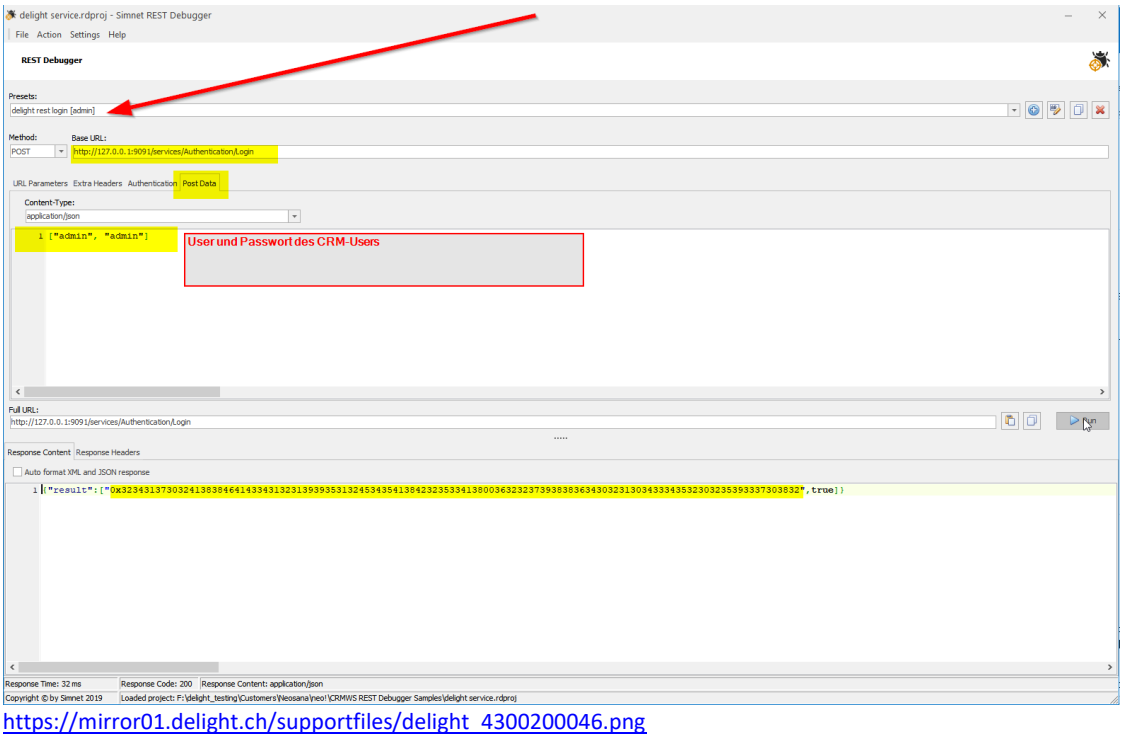
   Sobald der Client läuft, kann über localhost:9091 auf die REST-API zugegriffen werden.

   Stellen Sie bitte sicher, dass weder der Prozess (ML2Client.exe) noch der TCP/IP-Port (9091) von der lokalen Firewall gesperrt wird.

2. Beispiel Login-Request um SessionKey zu erhalten (Preset: «delight rest login [admin]»).

   Benutzername und Passwort werden als Parameter (JSON-Array) im HTTP-POST Request Body übergeben.

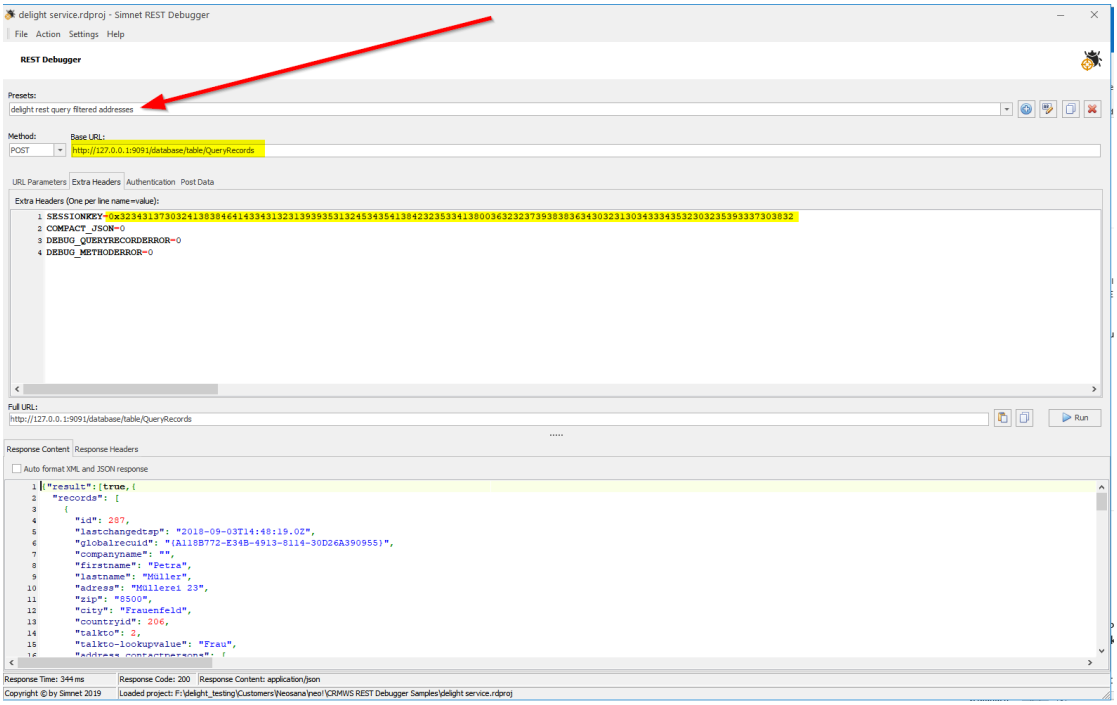
   Mit Klick auf «Senden» wird der Request ausgeführt und das Ergebnis im Reiter «Response Content» angezeigt. Beim Login-Request kommt als Response (result) die neue SessionKey zurück.



https://mirror01.delight.ch/supportfiles/delight_4300200046.png

3. Der in Schritt 2 erhaltenen SessionKey wird nun für alle weiteren (authentifizierten) Requests benötigt und muss im HTTP-Header «SESSIONKEY» mitgesendet werden.

   Beispiel: Irgend einen gewünschtes Preset auswählen, auf den Reiter «Extra Headers» wechseln und den SessionKey aus dem Login-Request im HTTP-Header "SESSIONKEY" einfügen. Mit Klick auf «Senden» wird der Request ausgeführt und das Ergebnis im Reiter «Response Content» angezeigt.

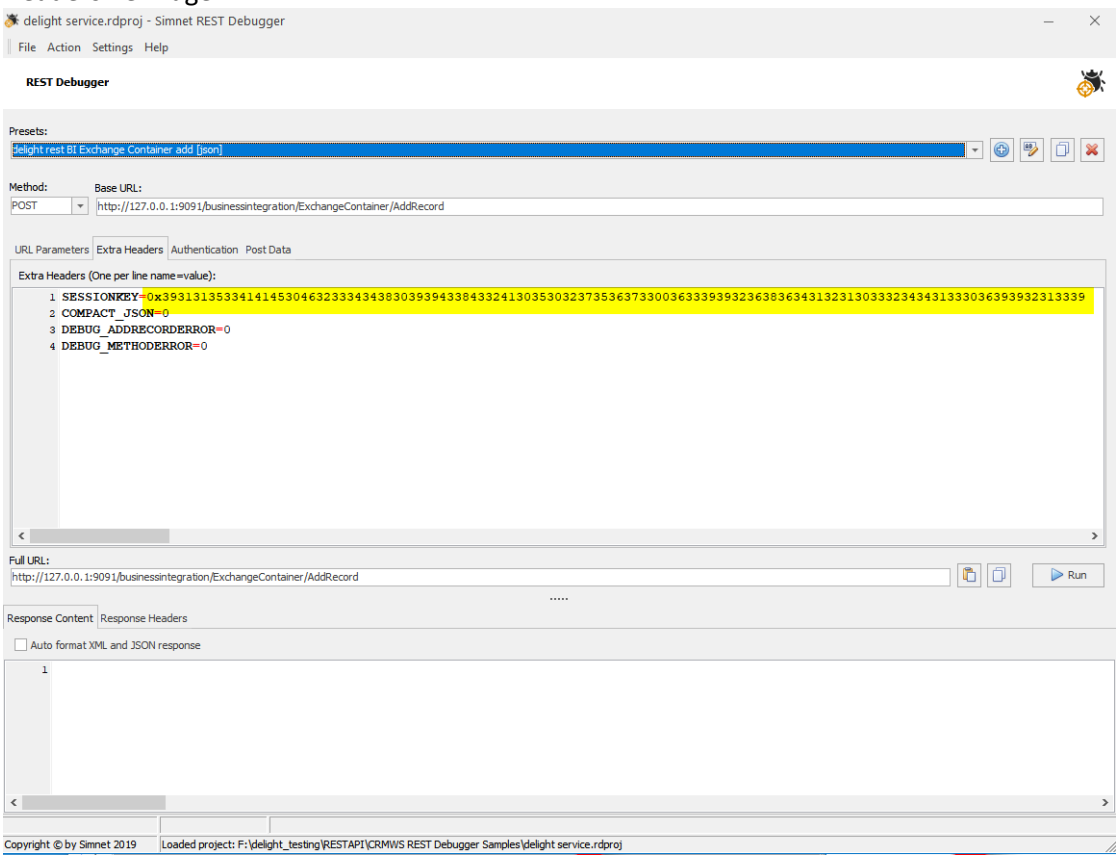


https://mirror01.delight.ch/supportfiles/delight_4600200128.png

Beispiel: Kontaktformular in generischen «BI Exchange Container» einliefern

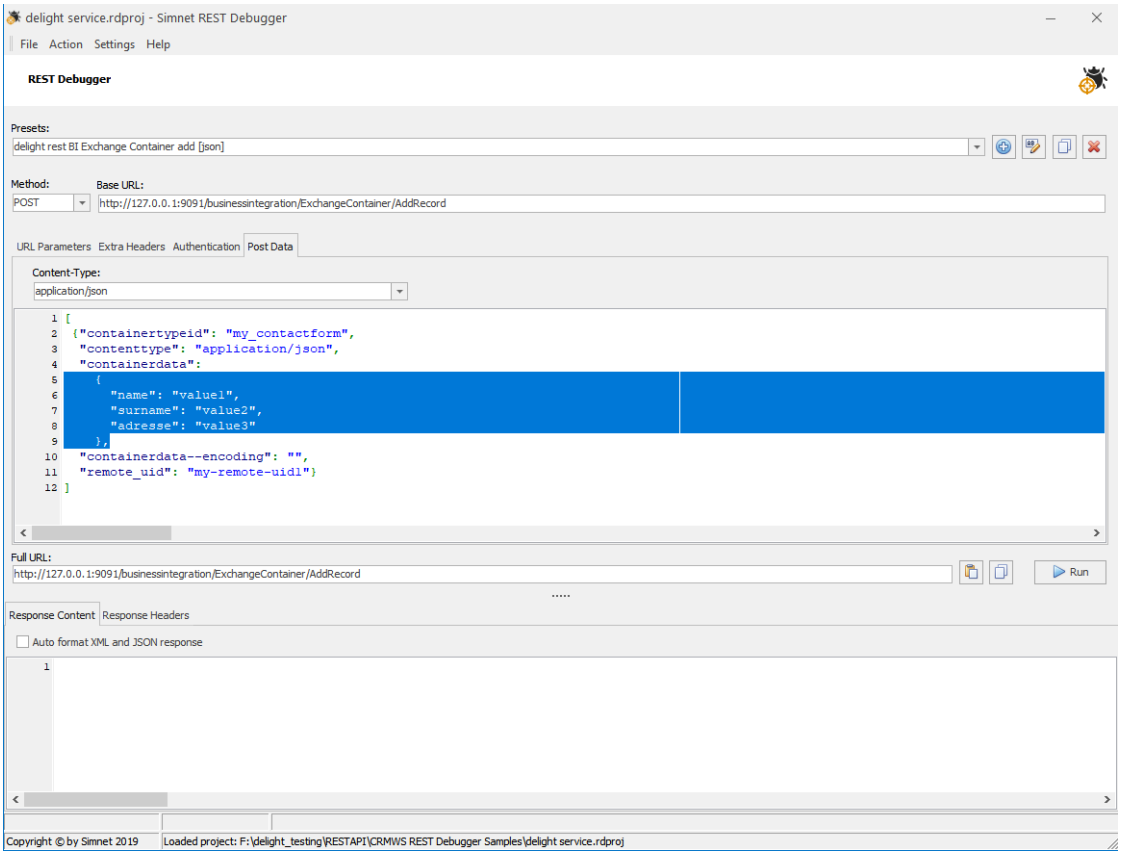
1. Login-Request ausführen um einen gültigen SessionKey zu erstellen, SessionKey kopieren.

2. Preset «delight rest BI Exchange Container add [json]» auswählen und SessionKey in «Extra Headers» einfügen



https://mirror01.delight.ch/supportfiles/delight_4600216201.png

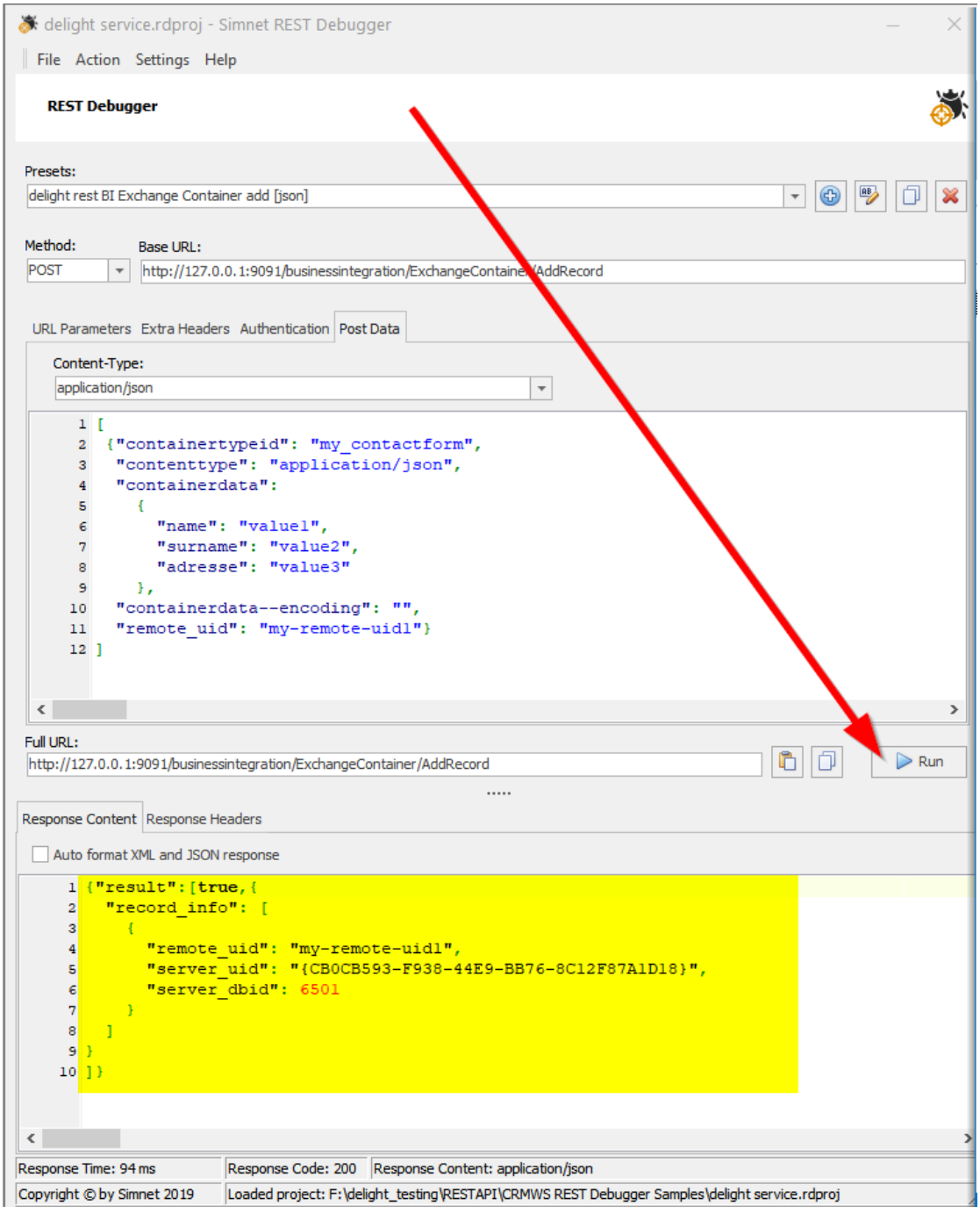
3. Im Reiter «Post Data» kann der Inhalt (in diesem Bsp. als JSON, es gibt auch Bsp. Presets für application/x-www-form-urlencoded) definiert werden.



https://mirror01.delight.ch/supportfiles/delight_4800216352.png

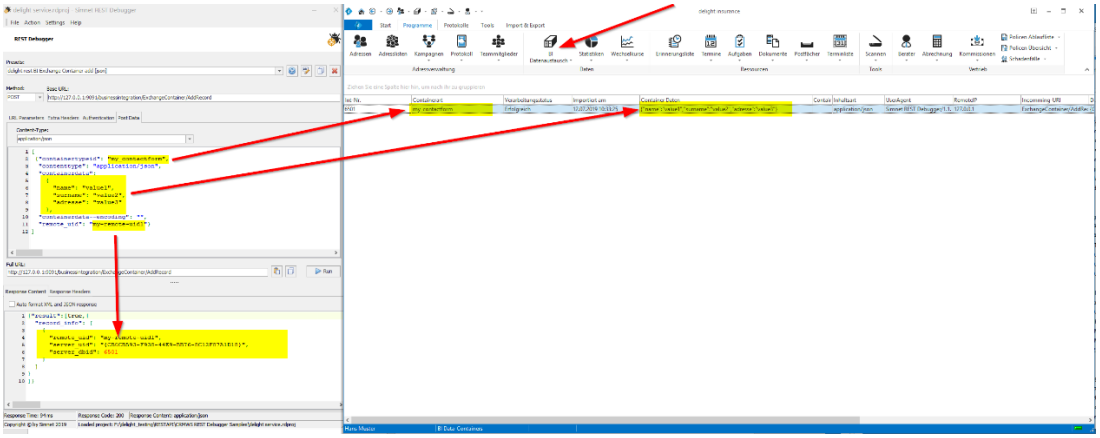
Anmerkung: Die Inhalte sind generisch. Die API validiert nur das Format (JSON) nicht aber die Inhalte. Die Inhalte können über eine Import-Konfiguration importiert werden. d. h. Struktur, Feldnamen, Wert u. s. w. sind grundsätzlich frei definierbar, da diese in der Importkonfiguration auf die delight-Felder gemappt werden können.

4. Mit Klick auf «Run» kann der Request ausgeführt werden. Im Reiter «Response Content» ist der Response der API ersichtlich.



https://mirror01.delight.ch/supportfiles/delight_5900216527.png

5. Der Request wird in delight unter «BI Datenaustausch» gespeichert. Die API speichert den Request in einem Container. Je nach Art des Containers werden die eingegangenen Daten mit unterschiedlichen Import-Konfigurationen im Hintergrund importiert

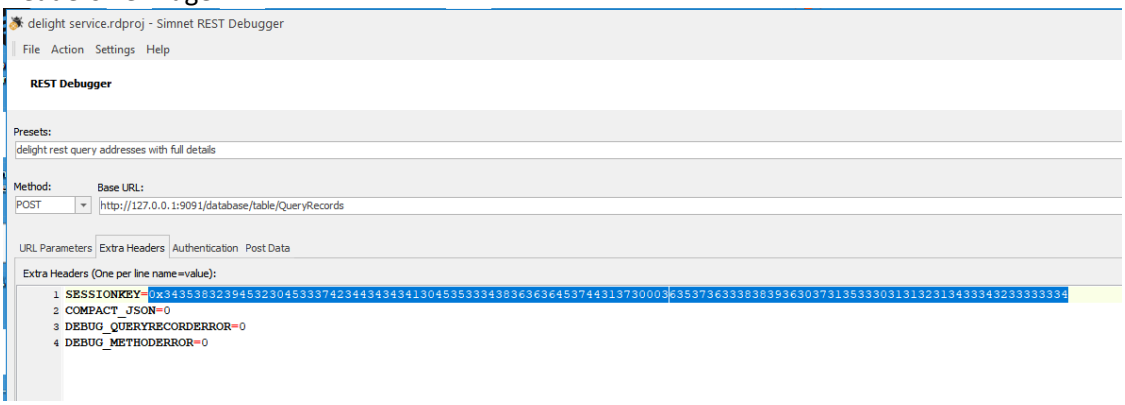


https://mirror01.delight.ch/supportfiles/delight_3400216402.png

Beispiel: Adressen mit aggregierten Detaildaten, gefiltert, abfragen

1. Login-Request ausführen um einen gültigen SessionKey zu erstellen, SessionKey kopieren.

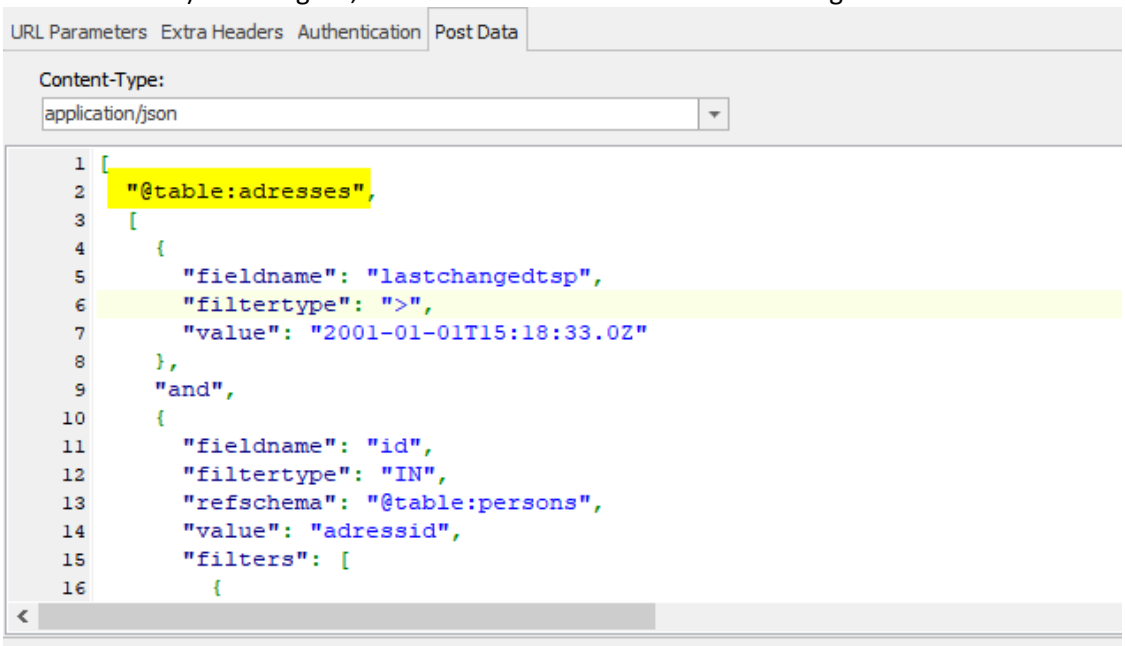
2. Preset «delight rest query addresses with full details» auswählen und SessionKey in «Extra Headers» einfügen.



https://mirror01.delight.ch/supportfiles/delight_3500237419.png

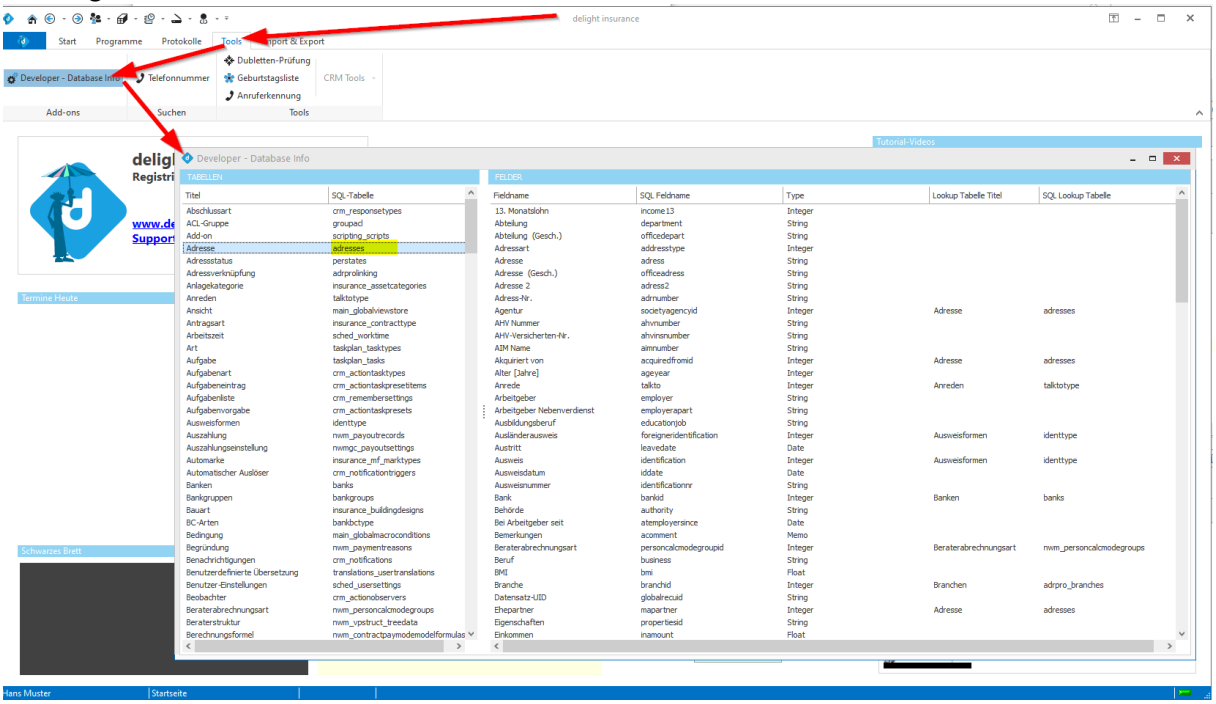
3. Im Reiter *Post Data* werden die Parameter (JSON-Array) für den REST-Call auf den Endpunkt /database/table/QueryRecords definiert.

4. **1. Parameter**: TableName aus der die Datensätze selektiert werden sollen. Alle in der jeweiligen Software-Fassung vorhanden Tabellen (je nach Modul, Ausführung, Customizing unterschiedlich) sind möglich, sofern der authentifizierte Benutzer Zugriff darauf hat.



https://mirror01.delight.ch/supportfiles/delight_4000237504.png

Anmerkung: Eine Liste aller verfügbaren Tabellen steht im «Developer - Database Info AddOn» zur Verfügung. Dieses AddOn kann auf jeder delight-Installation installiert werden und zeigt alle darin vorhanden Tabellen und Felder an.



https://mirror01.delight.ch/supportfiles/delight_5900237755.png

5. **2. Parameter**: Filter für die Daten.

https://mirror01.delight.ch/supportfiles/delight_4100237621.png

Anmerkung: Filter werden in einem strukturierten JSON abgebildet. Eine 1:1 «Übersetzung» in pseudo SQL, des in diesem Beispiel verwendeten Filters, würde folgenermassen aussehen:

```
SELECT * FROM adresses
WHERE (adresses.lastchangedtsp > "2001-01-01T15:18:33.0Z") and
adresses.id IN (SELECT persons.adressid FROM persons WHERE persons.ptypeid IN
(SELECT ptype.id FROM ptype WHERE ptype.descr = "Kunden" or ptype.descr = "Kunden A"
or ptype.descr = "Kunden B"))
```

6. **3. Parameter**: JSON-Array mit Felder und weiteren Einstellungen wie die jeweiligen Felder zurückgeliefert werden sollen.

https://mirror01.delight.ch/supportfiles/delight_1300237811.png

Anmerkung: Eine Liste aller verfügbaren Tabellen steht im «Developer - Database Info AddOn» zur Verfügung. Dieses AddOn kann auf jeder delight-Installation installiert werden und zeigt alle darin vorhanden Tabellen und Felder an.

7. **4. Parameter**: Einstellungen für Pagination.

https://mirror01.delight.ch/supportfiles/delight_1600237931.png

Anmerkung: In diesem Beispiel wir keine Pagination verwendet. In der Praxis solle für grössere Datenvolumen paginiert werden. Welche Paginierung Sinn macht, hängt sehr stark vom Inhalt ab. Die API kann grundsätzlich problemlos mehrere 1000 Datensätze in einem Request zurückliefern.

8. Mit Klick auf «Run» kann der Request ausgeführt werden. Im Reiter «Response Content» ist der Response der API ersichtlich.

https://mirror01.delight.ch/supportfiles/delight_2200238033.png

Fehlerhandling

Die REST-API gibt im Fehlerfall, wenn immer möglich, eine Strukturierte JSON-Fehlermeldung zurück. Ausnahmen können allgemeine HTTP-Fehler (z. B. Internal Server Error 500) sein, die bereits vor dem API-Layer auf dem Server entstehen.

Beispiel Fehlermeldung, wenn JSON im POST-Body fehlerhaft (nicht valid) ist


```
{
"errorCode":406,
"errorText":"sicSingle execution failed (probably due to bad input parameters) for
Table.QueryRecords"
}
```


Beispiel Fehlermeldung, wenn ein API-Aufruf inhaltlich fehlschlägt


```
{"result":[false,{
	"records": [],
	"recordset_info": {},
	"error": {
		"class": "Exception",
		"message": "Field \"\" is not a valid filterfield!"
	}
}
]}
```