



ESO9 JSON API

ESO9 international a.s.
U Mlýna 2305/22, 141 00 Praha 4 – Záběhlice
tel.: +420 228 809 000
e-mail: info@eso9.cz
www.eso9.cz

Zpracoval: Urych Tomáš
Dne: 20.2.2023

Revize: Urych Tomáš
Dne: 2.6.2025

Obsah

1.	ESO9 JSON API	2
2.	INSTALACE A KONFIGURACE API.....	2
2.1	PODMÍNKY PROVOZU.....	2
2.2	POSTUP INSTALACE	2
2.3	POSTUP KONFIGURACE.....	3
2.3.1	Přístup k databázi Windows účtem.....	3
2.3.2	Přístup k databázi SQL účtem.....	4
2.3.3	Licencování.....	4
2.4	PROVOZ API V KONTEJNERU	5
3.	ZÁKLADNÍ FUNKCE API.....	5
3.1	FORMÁT URL ADRESY	5
3.2	ZABEZPEČENÍ	5
3.3	FORMÁT DAT	6
3.4	FUNKCE GETDATA.....	6
3.5	FUNKCE GETDATAWITHCONDITION	7
3.6	FUNKCE GETROWSCOUNT	7
3.7	FUNKCE GETDATA PAGINATE	7
3.8	FUNKCE GETFILE	7
3.9	FUNKCE SETFILE	8
3.9.1	Nastavení oprávnění	8
3.9.2	Upload souboru.....	8
3.9.3	Automatizace volání s cURL	9
4.	NAPOJENÍ NA MS EXCEL	10
4.1	NAČTENÍ DAT Z WEBU.....	10
4.2	TRANSFORMACE DAT V POWER QUERY	11
4.3	PRÁCE S DATY V EXCELU.....	13

1. ESO9 JSON API

JSON API je obecné webové readonly rozhraní ESO9 určené pro získávání dat pro analytické účely/nástroje. Rozhraní poskytuje libovolná data z ESO9 přes web ve formátu JSON. Tento formát dat umí načítat např. Microsoft Excel, Power BI, Tableau a další.

API poskytuje webové rozhraní s několika základními funkcemi/URL. Pomocí přístupu k aplikační databázi načte požadovaná data a pošle je přes http(s) protokol klientovi ve formátu JSON.

2. Instalace a konfigurace API

Instalace a základní konfigurace API se týká pouze provozu ESO9 na prostředcích zákazníka (on-premise řešení). V rámci CLOUDu ESO9 řeší instalaci, konfiguraci a provoz API naše firma.

2.1 Podmínky provozu

API funguje jako samostatný web na webovém serveru IIS (Internet Information Server), který je součástí všech serverových edicí MS Windows Serveru. Instalace tedy počítá s existujícím a funkčním prostředím IIS.

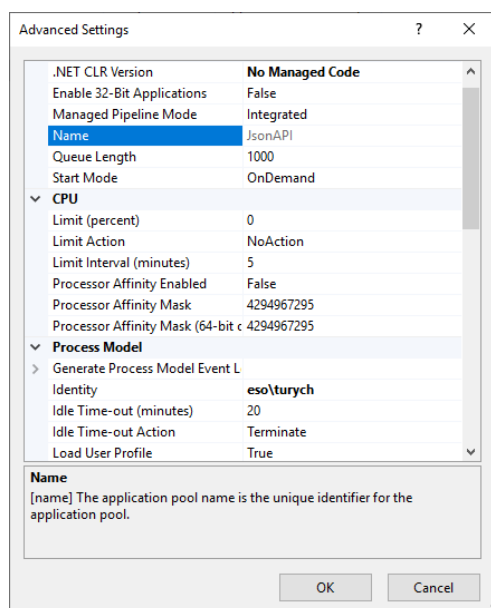
API je vytvořeno v prostředí .NET 8, potřebuje tedy ke svému běhu:

- RunTime .NET v8.0 (je součástí instalačního balíčku)
- ASP.NET Core Runtime Hosting Bundle v8.0 (<https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-8.0.15-windows-hosting-bundle-installer>)

2.2 Postup instalace

Obsah instalace je k dispozici na <https://podpora.eso9.cz/download/ESO9%20JSON%20API/NET%208/>.

- V IIS založíme nový aplikační pool. Klíčové vlastnosti budou:
 - *.NET CLR Version = No Managed Code*
 - *Managed pipeline mode = Integrated*
 - *Load User Profile = True*
 - *Identity = ApplicationPoolIdentity* (pokud chceme pro přístup API k aplikační databázi používat SQL autentikaci), *vyhrazený Windows účet* (pokud chceme pro přístup API k aplikační databázi používat Windows autentikaci)



- Pomocí instalačního programu nainstalujeme API. Jméno virtuální složky zvolíme libovolně; v dalším textu budeme předpokládat, že API je nainstalováno ve složce `c:\inetpub\wwwroot\JsonAPI`.
- Během instalace vybereme také aplikační pool; nejlépe ten, který jsme vytvořili o 2 kroky dříve.
- Autentikaci ponecháme webu anonymní (*Anonymous Authentication*).

2.3 Postup konfigurace

Veškerá konfigurace se týká souboru `appsettings.json` ve webu API (typicky `c:\inetpub\wwwroot\JsonAPI\appsettings.json`).

Konfigurační soubor obsahuje zejména:

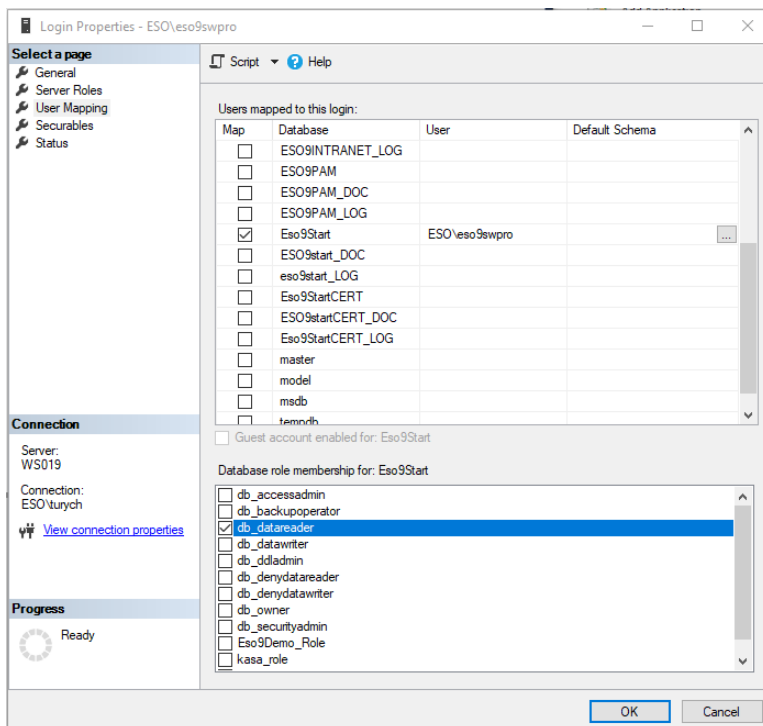
- definici připojení k aplikační databázi (tzv. connection string),
- přístupové údaje pro webový přístup k API,
- registrační číslo zákazníka pro získání licence.

Položky *Username* a *Password* obsahují přístupové údaje pro *Basic Authentication*, které bude API vyžadovat při každém přístupu k datům. Pokud je API vystavené do Internetu, měly by být jméno i heslo velmi silné.

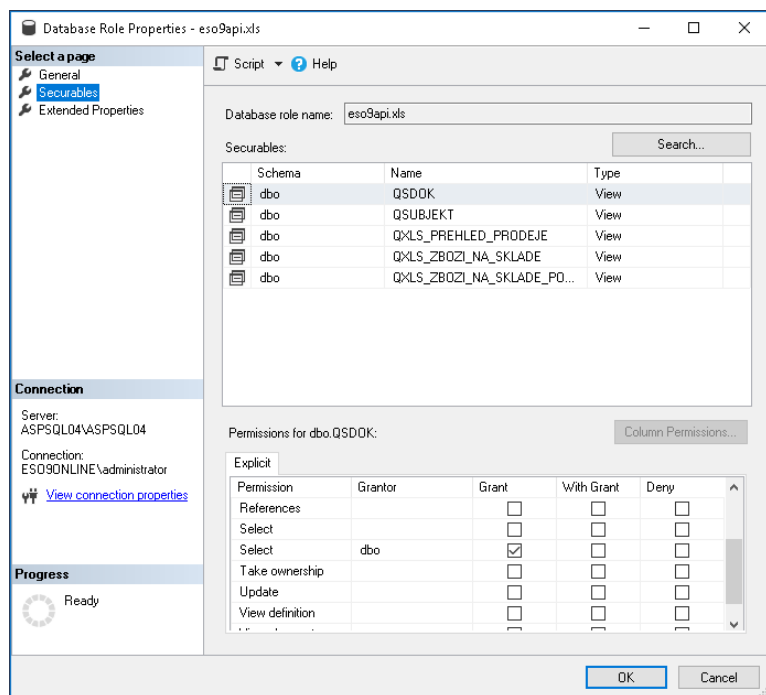
2.3.1 Přístup k databázi Windows účtem

Pokud plánujeme přistupovat k aplikační databázi přes Windows účet, použijeme v konfiguračním souboru formát *ConnectionStrings* – *AppDB*: „*Data Source=server;Initial Catalog=database;Integrated Security=SSPI;*“.

Na úrovni SQL Serveru nastavíme pro Windows účet (z předchozího odstavce) přístup pro čtení dat buď k celé databázi (tj. nastavíme účtu roli *db_datareader* pro celou aplikační databázi):



Nebo účet zařadíme do vyhrazené role, která má oprávnění pro čtení jen k vybraným databázovým objektům (tabulky, pohledy), z nichž budu přes API číst data (tj. nastavíme roli oprávnění *Select* na vybrané databázové objekty):



2.3.2 Přístup k databázi SQL účtem

Pokud plánujeme přistupovat k aplikační databázi přes Windows účet, použijeme v konfiguračním souboru formát *ConnectionStrings – AppDB*: „*Data Source=server;Initial Catalog=database;User Id=xxxx;Password=xxxx;*”.

Nastavení oprávnění vyhrazeného SQL účtu je řešeno analogicky s nastavením Windows účtu.

2.3.3 Licencování

Licencování se týká pouze zákazníků s vlastními servery, pro zákazníky na našem CLOUDu je API k dispozici zdarma.

Při provozu na vlastních serverech vyžaduje každá instance API jednu licenci. Pokud tedy na jednom webovém serveru poběží tři instance API, z nichž každá přistupuje k jiné databázi, je třeba mít zakoupené tři licence.

Licencování probíhá stejným způsobem, jako licencování běžných aplikací ESO9 nebo jejich doplňků, tj. online pomocí webové služby. K tomu je zapotřebí zadat do konfiguračního souboru registrační číslo zákazníka do položky *regNumber*. Pokud je API nainstalované na stejném webovém serveru, na němž již běží aplikační server ESO9, není zpravidla třeba tento server (resp. jeho HW otisk / aktivační klíč) znovu registrovat oproti licenční službě. Pokud se z daného serveru nikdy licence ESO9 neaktivovala (běžně se to dělá ze *Správce ESO9*), je třeba registraci aktivačního klíče serveru jednorázově provést.

2.3.3.1 Registrace aktivačního klíče

Stejně, jako se při aktivaci licence aplikací ESO9 registruje HW otisk daného aplikačního serveru, je třeba zaregistrovat i webový server, na němž je provozováno JSON API. Registrace se provádí z příkazového řádku pomocí programu *ESO9ExcelAPI.exe*, který najdeme v instalačním adresáři API:

```
ESO9ExcelAPI.exe /register
```

Výsledek registrace je následně uložen v logovacím souboru v adresáři *Logs*. Při úspěšné registraci bude v logovacím souboru záznam:

```
2024-03-12T16:49:36.9426275+01:00 [INF] Registering FP. (3971d2d2)
2024-03-12T16:49:39.9395043+01:00 [INF] FP registered successfully. (4bcca35e)
```

Při neúspěšné registraci bude v logovacím souboru záznam:

```
2024-03-12T16:51:43.5466877+01:00 [INF] Registering FP. (3971d2d2)
2024-03-12T16:51:46.9969519+01:00 [ERR] HW fingerprint not registred!
On-line licencování je aktivováno pro jiný aplikační server, než aktuální.
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX - uložený aktivační klíč.
YYYY-YYYY-YYYY-YYYY-YYYY-YYYY - aktuální aktivační klíč.
Došlo k přesunu na jiný server nebo k upgrade serveru.
Zažádejte o změnu aktivace.
Nalezen zákazník r.č. "xxxxxx" Testovací subjekt (cb3ae8ed)
```

Pokud se registrace nepodaří, je třeba požádat na Podpoře ESO9 o povolené registrace nového serveru, popř. rovnou zaslat aktuální aktivační klíč z logovacího souboru.

2.3.3.2 Kontrola funkční licence

Při startu aplikačního poolu v IIS se kontroluje licence pro danou databázi. Pokud je licence v pořádku, objeví se v logovacím souboru záznam:

```
2024-03-12T16:31:45.6360734+01:00 [INF] Checking licence. (d66002f5)
2024-03-12T16:31:47.5331903+01:00 [INF] Licence checked successfully. (146308cb)
```

Pokud licence dostupná nebude, v logovacím souboru bude záznam:

```
2024-03-12T16:09:03.9168193+01:00 [INF] Checking licence. (d66002f5)
2024-03-12T16:09:05.7161384+01:00 [ERR] Not licenced!
```

Nepodaří-li se úspěšně zalicencovat API, kontaktujte prosím Podporu ESO9.

2.4 Provoz API v kontejneru

JSON API lze provozovat (kromě samostatné instalace, viz výše) i v Docker kontejneru. Výhodami kontejnerizace jsou zejména:

- Izolace prostředí, tj. nezávislost na IS a verzích knihoven. Aktuální verze API je připravena na provoz v Dockeru pod Windows i pod Linuxem
- Snadná a rychlá instalace i aktualizace – stačí stáhnout image. Základní image pro obě prostředí (Windows i Linux) je k dispozici na vyžádání v naší společnosti.
- Jednoduché škálování = další instanci API spustím z existujícího image v řádu sekund.
- Snadná migrace mezi servery = přenositelný kontejner.
- Automatizace nasazení.
- Jednoduché zálohování a obnova (práce s image a snapshoty).

3. Základní funkce API

3.1 Formát URL adresy

Všechny funkce API mají formát URL adresy. Adresa sestává z fixní části a z části s parametry. Parametry pak určují, jaká data se budou z ESO9 získávat, popř. dále upřesňují jejich rozsah.

V případě, že se v URL parametrech vyskytnou nepřípustné znaky, je třeba je převést tzv. URL encodingem. Viz např. <https://www.urlencoder.org/>.

3.2 Zabezpečení

Všechny funkce API (tj. URL adresy) jsou zabezpečeny *Basic Authentication*. Jméno a heslo se nastavují v konfiguračním souboru, viz kapitola 2.3.

3.3 Formát dat

Všechny funkce API vrací data ve formátu *JSON*. Tento formát umí zpracovat všechny nástroje pro datovou analýzu či vizualizaci, ale i např. databázové systémy typu *MS SQL Server*.

Kořenovým elementem je vždy název požadované tabulky nebo pohledu. Jednotlivé záznamy jsou ve formátu jednorozměrného pole.

Příklad: (první 2 záznamy z tabulky SUBJEKT)

```
{
  "subjekt": [
    {
      "IDSUBJEKT": 208,
      "KOD_SUBJEKTU": "0028",
      "ICO": "00006912",
      "DIC": "CZ00006912",
      "SUBJ_NAZEV": "Finanční úřad",
      "IDSUBJADR": 295,
      "IDSUBJOSOBA": 323,
      "IDSUBJBUCET": 160,
      "VLSTAVSUBJEKT": 0,
      "VLSUBJ_KONSOLIDACE": 0,
      "VLSUBJTYPSPOL": 0,
      "VLSUBJOB RAT": 0,
      "VLSUBJPOCETZAM": 0,
      "VLPRAVNIFORMA": 14,
      "IDUZIVATEL_UPDATE": 327,
      "DTUPDATE": "2018-12-17T14:53:37",
      "VLPLATCEDPH": 0,
      "VLNEDUVERYHODNYPLATCE": -1
    },
    {
      "IDSUBJEKT": 219,
      "KOD_SUBJEKTU": "0001",
      "ICO": "27624609",
      "DIC": "CZ27624609",
      "SUBJ_NAZEV": "ESO9 international a.s.",
      "IDSUBJADR": 365,
      "IDSUBJOSOBA": 335,
      "IDSUBJBUCET": 188,
      "SUBJ_POZNAMKA": "Informační systémy ESO9",
      "VLSTAVSUBJEKT": 0,
      "VLSUBJ_KONSOLIDACE": 0,
      "VLSUBJTYPSPOL": 0,
      "VLSUBJOB RAT": 0,
      "VLSUBJPOCETZAM": 0,
      "SUBJEKT_WEBSITE": "www.eso9.cz",
      "VLPRAVNIFORMA": 1,
      "VLEDIPROVIDER": 1,
      "VLEDIFORMAT": 2,
      "DATOVASCHRANKAID": "czxfpqh",
      "IDOBECIS_TYPDS": 41,
      "IDUZIVATEL_UPDATE": 323,
      "DTUPDATE": "2023-01-12T10:25:01",
      "CISLODU": "",
      "VLPLATCEDPH": 2,
      "VLNEDUVERYHODNYPLATCE": 0
    }
  ]
}
```

3.4 Funkce getData

Vrací všechna data z dané tabulky nebo pohledu bez omezení.

URL: <https://{Server}/JsonAPI/Data/getData/{DataSource}>

Parametr {Server}: jméno webového serveru, který hostuje API.

Parametr {DataSource}: jméno tabulky nebo pohledu z ESO9, jehož data funkce vrátí. Pokud používáme pro uživatelský účet přistupující k aplikační databázi oprávnění pro čtení, není třeba nic dále nastavovat. Pokud používáme pro uživatelský účet přistupující k aplikační databázi oprávnění pouze pro vybrané databázové objekty, je třeba každému takovému objektu (tabulce, pohledu) nastavit oprávnění *Select*.

Příklad: <https://api.eso9.cz/JsonAPI/Data/getData/QSUBJEKT>

Příklad vrátí všechna data z pohledu QSUBJEKT, tj. informace o subjektech (obchodních partnerech).

3.5 Funkce getDataWithCondition

Vrací data z dané tabulky nebo pohledu omezená zadanou podmínkou.

URL: <https://{Server}/JsonAPI/Data/getDataWithCondition/{DataSource}?{Condition}>

Parametr {DataSource}: jméno tabulky nebo pohledu z ESO9, jehož data funkce vrátí.

Parametr {Condition}: podmínka aplikovaná na tabulku/pohled. Podmínka se uvádí ve tvaru *condition=SQL_where_condition*, tj. používá se syntaxe Where podmínek z jazyka T-SQL.

Příklad:

https://api.eso9.cz/JsonAPI/Data/getDataWithCondition/QHDOK_FAV?condition=UCET_OBD%3E%272020.01%27

Příklad vrátí data z pohledu QHDOK_FAV (tj. vydané faktury) omezená na účetní období větší, než 2020.01 (UCET_OBD>'2020.01'). Nepřípustné znaky v URL jsou převedené URL encodingem (viz kapitola 3.1).

3.6 Funkce getRowCount

Vrací počet řádků tabulky nebo pohledu.

URL: <https://{Server}/JsonAPI/Data/getRowCount/{DataSource}>

Parametr {DataSource}: jméno tabulky nebo pohledu z ESO9, počet jehož řádků funkce vrátí.

Příklad: <https://api.eso9.cz/JsonAPI/Data/getRowCount/PSC>

Příklad vrátí počet řádků tabulky s PSČ.

3.7 Funkce getDataPaginate

Vrací stránkovaná data z tabulky nebo pohledu. Stránku s daty určuje její počátek a velikost. Funkce je určena pro postupné načítání rozsáhlých dat. Pro provoz v kontejneru jediný způsob, jak postupně načíst rozsáhlejší data.

URL: <https://{Server}/JsonAPI/Data/getDataPaginate/{DataSource}?fromRow=x&rows=y>

Parametr {DataSource}: jméno tabulky nebo pohledu z ESO9, jehož data funkce vrátí.

Parametr fromRow: číslo řádku, odkud se mají data číst.

Parametr rows: počet řádků, které má funkce vrátit.

Příklad: <https://api.eso9.cz/JsonAPI/Data/getDataPaginate/PSC?fromRow=2000&rows=300>

Příklad vrátí 300 řádků tabulky s PSČ počínaje řádkem č.2000.

3.8 Funkce getFile

Vrací soubor z DMS na základě jeho unikátního identifikátoru GUID.

URL: <https://{Server}}/JsonAPI/File/getFile/{fileGUID}>

Parametr {fileGUID}: identifikátor (GUID) souboru z DMS ESO9, který chceme stáhnout.

Příklad: <https://api.eso9.cz/JsonAPI/File/getFile/44C985A2-48F1-4DE0-8B2C-8CFD314E1747>

Příklad vrátí soubor s daným GUID identifikátorem.

3.9 Funkce setFile

Uloží soubor do DMS a spustí nad ním uživatelskou proceduru. Na rozdíl od všech předchozích funkcí se nejedná o čtení dat (HTTP GET), ale o jejich zápis (HTTP POST).

Soubor se zakládá pouze do dokumentové databáze do tabulky *ESO9DOC*; k souboru se tedy nezakládá záznam do aplikační databáze (do navázané tabulky *DOKUMENT*). Mechanismus je určen primárně pro jednoduché souborové importy, kdy po zpracování obsahu souboru tento zrušíme.

3.9.1 Nastavení oprávnění

Tato kapitola je rozšířením kapitoly 2.3.1 pro účely volání funkce *setFile*.

Protože funkce zapisuje do DMS ESO9, je třeba Windows účtu (resp. SQL účtu), pod nímž API přistupuje k dokumentové databázi, nastavit minimálně role *db_datareader* a *db_datawriter*.

Pokud se po uložení souboru volá uživatelská procedura, je třeba příslušné roli nastavit k této proceduře oprávnění *Execute*.

3.9.2 Upload souboru

Pro upload souboru do DMS se používá tzv. formát *MultiPart Form Data*. Ten spočívá v odeslání vlastního souboru v těle HTTP requestu, zatímco případné další parametry se posílají na URL (v našem případě uživatelská procedura). V těle requestu je povinná jediná MultiPart část s názvem „file“, viz také příklad níže.

URL: <https://{Server}}/JsonAPI/File/setFile?userDBProcedure={spProcedure}>

Parametr {spProcedure}: jméno uživatelské procedury, která se zavolá po úspěšném uložení souboru do DMS ESO9. Jediným vstupním parametrem procedury je GUID založeného souboru. Procedura je nepovinná.

Příklad uživatelské procedury:

```
/*
    Procedura volaná PO importu souboru do DMS přes JSON API
*/
CREATE OR ALTER PROCEDURE dbo.spUploadFile
    @FILEGUID uniqueidentifier
AS
    Set Nocount ON
    -- výkonná část procedury
    RETURN 0
GO
```

Příklad HTTP requestu:

```

POST /Data/setFile?userDBProcedure=spUploadFile HTTP/1.1
Authorization: Basic RVNPOUV4Y2VsQVBjOklhc2xvMTIz
User-Agent: PostmanRuntime/7.33.0
Accept: */*
Postman-Token: af28f84c-312a-4572-a989-131e8cd34018
Host: localhost:7250
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: multipart/form-data; boundary=-----482225562923024807805787
Content-Length: 272161

-----482225562923024807805787
Content-Disposition: form-data; name="file"; filename="Invoice2021_346.pdf"
<# Invoice2021_346.pdf>
-----482225562923024807805787--

```

Struktura MultiPart requestů je popsána např. na <https://swagger.io/docs/specification/describing-request-body/multipart-requests/>.

3.9.3 Automatizace volání s cURL

cURL je řádková utilita pro přenos dat z/na server pomocí protokolu HTTP (ve skutečnosti podporuje celou řadu dalších protokolů). Lze ji stáhnout např. z <https://curl.se/download.html>.

Pomocí tohoto nástroje lze jednoduše zavolat jakékoli webové API včetně všech zde uvedených metod.

Nicméně u metod typu GET s výstupem ve formátu JSON je stále lepší použít jako klienta MS Excel.

Pokud tedy klientský software (např. e-shop) umí volat *HTTP API*, lze soubory přenášet přímo z něj. Pokud je třeba manuálně přenášet a importovat soubory ze souborového systému, lze to automatizovat právě pomocí nástroje *cURL*.

3.9.3.1 Příklad s importem souboru

Příklad uploadu a importu souboru z příkazové řádky (volá API funkci *setFile* na serveru *Win2019*, po uploadu souboru *API.pdf* spustí uživatelskou proceduru *spUploadFile*):

```

curl -X POST "https://win2019/ESO9JsonAPI/File/setFile?userDBProcedure=spUploadFile"
--user ESO9ExcelAPI:Heslo123
-H "accept: */*"
-H "Content-Type: multipart/form-data"
-F "file=@API.pdf;type=application/pdf"
-insecure

```

3.9.3.2 Příklad s hromadným stahováním souborů z DMS

Příklad na automatické stažení více souborů z DMS z příkazové řádky pomocí skriptu v Powershellu.

Předpokládejme, že seznam souborů ke stažení máme definovaný v pohledu *QDOKUMENTOBR*. Skript tak nejprve získá pomocí tohoto pohledu seznam souborů určených ke stažení a následně je v cyklu stáhne.

```

#####
# Stažení souborů z příkazové řádky z aplikace ESO9 pomocí JSON API
#####

# Definice proměnných
$username = "ESO9ExcelAPI"
$password = "Heslo123"
$baseDirectory = "c:\Temp\curl-8.3.0_1-win64-mingw\bin\"
$outputFile = "$baseDirectory\FILE.jpg"

# Dočasné umístění pro JSON soubor s odpovědí ze serveru
$jsonFile = "$baseDirectory\response.json"

# Stažení JSON dat

```

```
$url = "https://win2019-tu2/ESO9JSONAPI/Data/getData/QDOKUMENTOBR"
$curlCommand = "curl.exe -X GET \"$url\" --user \"$username:$password\" --header `\"Accept: */*\" -o \"$jsonFile\" --insecure\"
Invoke-Expression $curlCommand

# Čtení a parsování JSON souboru
$jsonData = Get-Content -Raw -Path $jsonFile | ConvertFrom-Json

# Cyklus přes všechny soubory ke stažení
$url = "https://win2019-tu2/ESO9JSONAPI/File/getFile"

foreach ($doc in $jsonData.QDOKUMENTOBR) {
    $fileGuid = $doc.DOKUMENTFILEGUID
    $fileName = $doc.FILENAME

    # Sestavení URL pro stažení jednoho souboru
    $downloadUrl = "$url/$fileGuid"

    # Cesta k uložení jednoho souboru
    $outputFile = "$baseDirectory\$fileName"

    # Stažení souboru pomocí cURL
    $curlDownloadCommand = "curl.exe -X GET \"$downloadUrl\" --user \"$username:$password\" --header `\"Accept: */*\" -o \"$outputFile\" --insecure\"
    Write-Output "File URL: $downloadUrl"
    Invoke-Expression $curlDownloadCommand

    Write-Output "Downloaded: $fileName"
}
```

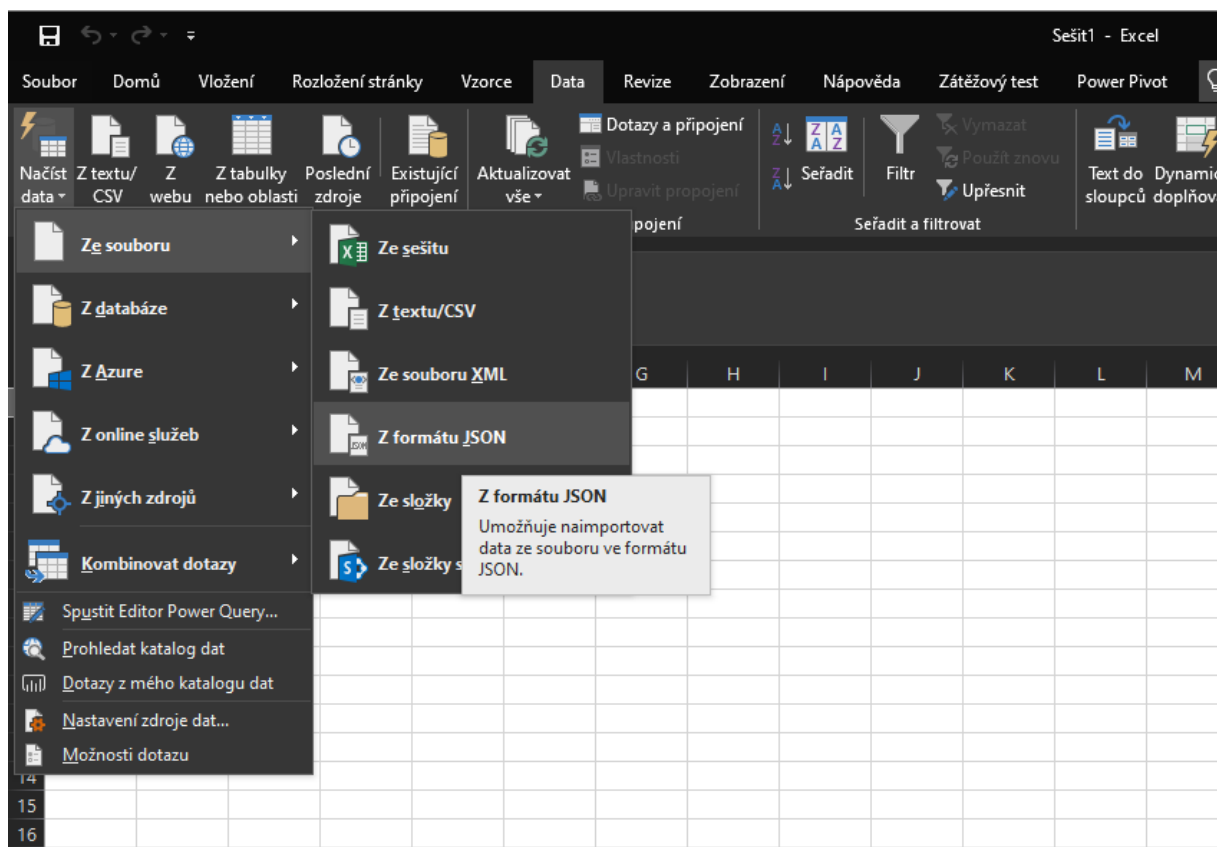
4. Napojení na MS Excel

Protože MS Excel patří mezi nejjednodušší nástroje pro datovou analýzu, předvedeme si způsob napojení na webové API právě na něm.

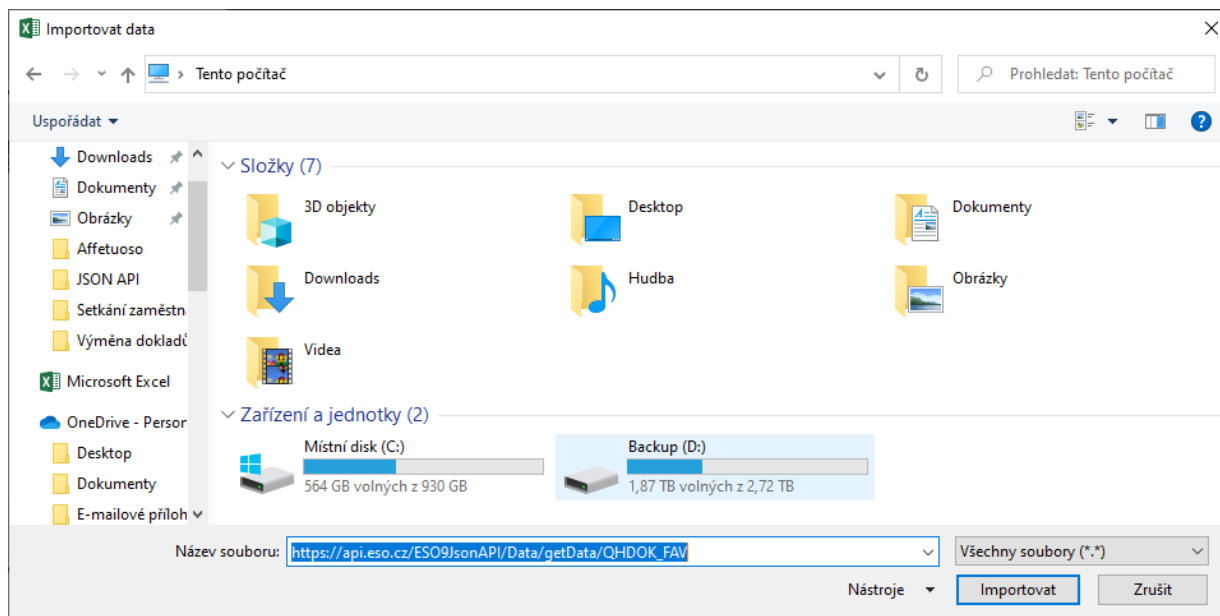
Předpokládejme, že máme zprovozněné API pro pohled QHDOK_FAV (seznam dokladů vydaných faktur) na adrese: https://api.eso.cz/ESO9JsonAPI/Data/getData/QHDOK_FAV.

4.1 Načtení dat z webu

Ačkoli se jedná o načtení dat z webového rozhraní, nejjednodušší cestou je zvolit v menu *Data – Načíst data – Ze souboru – Z formátu JSON*:



Do položky **Název souboru** zadáme URL adresu rozhraní a po upřesnění ještě jméno a heslo pro přístup k webovému API:

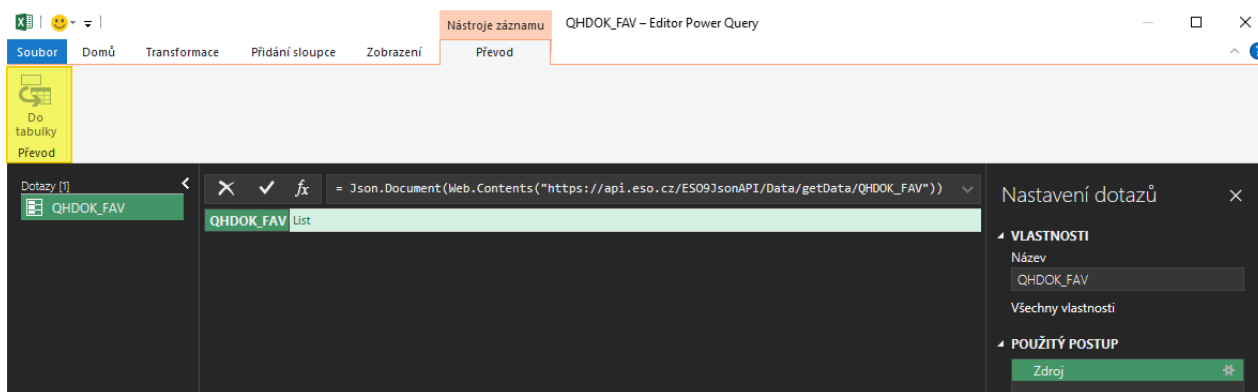


Po zadání přístupových údajů se načtou data do nástroje *Editor Power Query*.

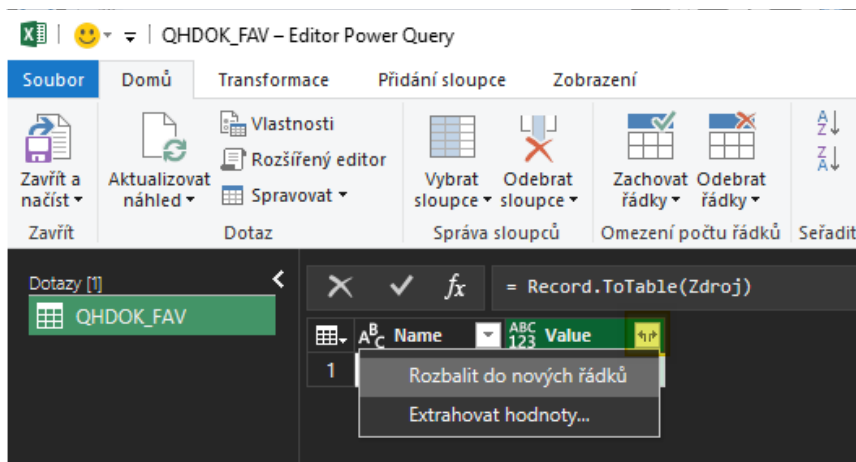
4.2 Transformace dat v Power Query

Data načtená z webového rozhraní jsou ve formátu JSON a při prvním napojení je musíme transformovat na tabulku (tj. strukturu řádky / sloupce).

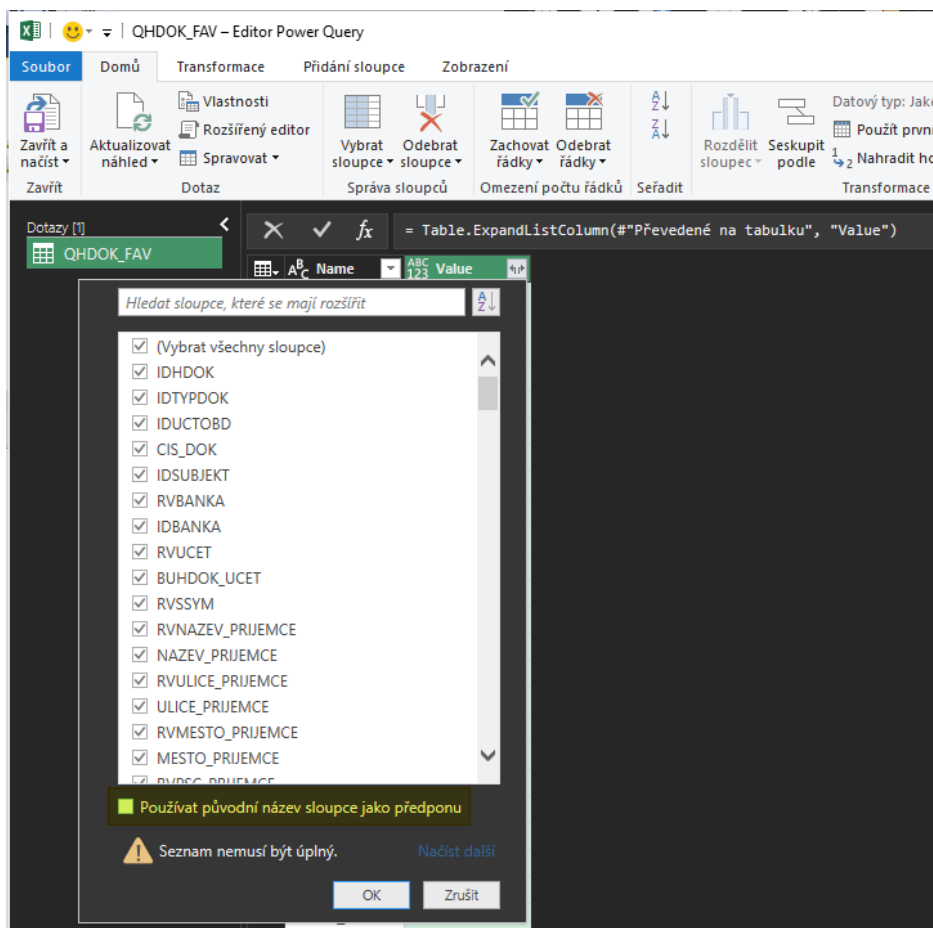
Nejprve zvolíme převod dat do tabulky:



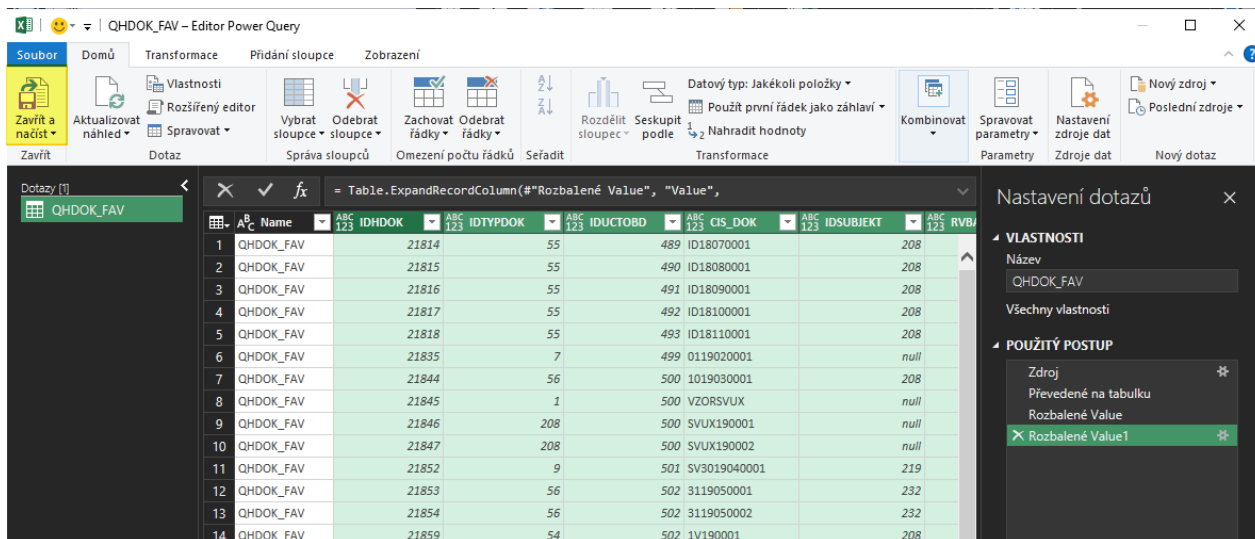
Data se načtou do tabulky, ovšem všechna jen do jednoho řádku. Pro transformaci na reálnou tabulkovou strukturu použijeme volbu *Rozbalit do nových řádků*:



Získáme tím tabulku, jejíž všechny sloupce se jmenují dle zdrojového pohledu (v našem příkladu tedy *QHDOK_FAV*). Pro úpravu názvů tabulky odškrtneme volbu *Používat původní název sloupce jako předponu*:



Tímto krokem získáme původní tabulkovou strukturu i se správně pojmenovanými sloupci. Stiskem tlačítka **Zavřít a načíst** se vrátíme zpět do Excelu:



4.3 Práce s daty v Excelu

Návratem do Excelu se dostaneme ke všem načteným datům. Stiskem tlačítka **Aktualizovat** si kdykoli můžeme načtená data obnovit online a to již bez jakýchkoli dotazů a datových transformací.

	A	B	C	D	E	F	G	H	I	J	K
	Name	IDHDOK	IDTYPDOK	IDUCTOBD	CIS_DOK	IDSUBJEKT	RVBANKA	IDBANKA	RVUCET	BUHDOK_UCET	RVSSYM
1	QHDOK_FAV	21814	55	489	ID18070001	208	0	141	0	19-7487627041	
2	QHDOK_FAV	21815	55	490	ID18080001	208	0	141	0	19-7487627041	
3	QHDOK_FAV	21816	55	491	ID18090001	208	0	141	0	19-7487627041	
4	QHDOK_FAV	21817	55	492	ID18100001	208	0	141	0	19-7487627041	
5	QHDOK_FAV	21818	55	493	ID18110001	208	0	141	0	19-7487627041	
6	QHDOK_FAV	21835	7	499	0119020001		0		0		
7	QHDOK_FAV	21844	56	500	1019030001	208	0	141	0	19-7487627041	
8	QHDOK_FAV	21845	1	500	VZORSVUX		0		0		
9	QHDOK_FAV	21846	208	500	SVUX190001		0		0		
10	QHDOK_FAV	21847	208	500	SVUX190002		0		0		
11	QHDOK_FAV	21852	9	501	SV3019040001	219	0	2	0	2509840204	
12	QHDOK_FAV	21853	56	502	3119050001	232	0	4	0	1303998319	
13	QHDOK_FAV	21854	56	502	3119050002	232	0	4	0	1303998319	
14	QHDOK_FAV	21859	54	502	1V190001	208	0	141	0	19-7487627041	
15	QHDOK_FAV	21863	1	502	VZORDZA		0		0		
16	QHDOK_FAV	21864	209	502	DZ19050001		0		0		
17	QHDOK_FAV	21865	113	502	ZD19050001		0		0		
18	QHDOK_FAV	21866	55	494	ID18120001	208	0	141	0	19-7487627041	
19	QHDOK_FAV	21867	55	498	ID19010001	208	0	141	0	19-7487627041	
20	QHDOK_FAV	21868	55	499	ID19020001	208	0	141	0	19-7487627041	
21	QHDOK_FAV	21869	55	500	ID19030001	208	0	141	0	19-7487627041	
22	QHDOK_FAV										

Pokud Excelový sešit zpřístupním komukoli jinému, bude mít k dispozici naposledy načtená data. Při požadavku na jejich obnovu se Excel zeptá na zadání přístupového jména a hesla k API. Po jeho zadání na jiném PC si Excel přístupové údaje uloží a při příští aktualizaci se již neptá.