



# **ESO9 JSON API**

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

Urych Tomáš 20.2.2023

Revize: Urych Tomáš Dne: 2.6.2025

## Obsah

1.		ESO9	JSON API	2
2.		INSTA	LACE A KONFIGURACE API	2
2	.1	Podn	IÍNKY PROVOZU	2
2	.2	Ροςτι	JP INSTALACE	2
2	.3	Ροςτι	JP 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	PROV	oz API v kontejneru	5
3.		ZÁKLA	ADNÍ FUNKCE API	5
3	.1	Form	ÁT URL ADRESY	5
3	.2	ZABEZ	PEČENÍ	5
3	.3	Form	ÁT DAT	6
3	.4	FUNK	CE GETDATA	6
3	.5	FUNK	CE GETDATAWITHCONDITION	7
3	.6	FUNK	CE GETROWSCOUNT	7
3	.7	FUNK	CE GETDATAPAGINATE	7
3	.8	FUNK	CE GETFILE	7
3	.9	FUNK	CE SETFILE	8
	3.9	.1	Nastavení oprávnění	8
	3.9	.2	Upload souboru	8
	3.9	.3	Automatizace volání s cURL	9
4.		NAPO	JENÍ NA MS EXCEL	0
4	.1	Načti	ENÍ DAT Z WEBU	0
4	.2	TRANS	SFORMACE DAT V POWER QUERY	1
4	.3	Práce	es daty v Excelu	3

# 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 (<u>https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-8.0.15-windows-hosting-bundle-installer</u>)

## 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)

	.NET CLR Version	No Managed Code	^					
	Enable 32-Bit Applications	False						
	Managed Pipeline Mode	Integrated						
	Name	JsonAPI						
	Queue Length	1000						
	Start Mode	OnDemand						
~	CPU							
	Limit (percent)	0						
	Limit Action	NoAction						
	Limit Interval (minutes)	5						
	Processor Affinity Enabled	False						
	Processor Affinity Mask	4294967295						
	Processor Affinity Mask (64-bit c 4294967295							
~	Process Model							
>	Generate Process Model Event	t L						
	Identity	eso\turych						
	Idle Time-out (minutes)	20						
	Idle Time-out Action	Terminate						
	Load User Profile	True	$\sim$					

- 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):

Select a page	Script	🔻 😯 Help								
Server Roles	Users mapped to this login:									
Securables	Мар	Database	User	Default Schema	^					
Status		ESO9INTRANET_LOG								
		ESO9PAM								
		ESO9PAM_DOC								
		ESO9PAM_LOG								
		Eso9Start	ESO\eso9swpro							
		ESO9start_DOC								
		eso9start_LOG								
		Eso9StartCERT								
		ESO9startCERT_DOC								
		Eso9StartCERT_LOG								
		master								
		model								
		msdb			_					
Connection		tempdh			×					
	Guest	account enabled for: Eso99	itart							
Server: WS019										
Constraint	Database	e role membership for: Eso95	tart							
ESO\turvch	db_a	ccessadmin			^					
······································		ackupoperator atareader			_					
view connection properties	db d	atawriter								
	db_d	dladmin								
	dbd	enydatareader								
Or outrace		enydatawriter								
Togrooo	db_securityadmin									
Ready	Eso9	Demo_Role								
AT A A A A A A A A A A A A A A A A A A	kasa	role			~					

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):

🗑 Database Role Properties - e	so9api.xls				- 0	×
Select a page	🖵 Script 👻 😯 Help					
Securables						
Extended Properties	Database role name:					
	Securables:				Search	
	Schema	Name		Туре		
	dbo	QSDOK		View		
	dbo	QSUBJEK	(T	View		
	🗇 dbo	QXLS_PF	REHLED_PRODEJE	View		
	🗇 dbo	QXLS_ZE	OZI_NA_SKLADE	View		
	🗇 dbo	QXLS_ZE	OZI_NA_SKLADE_	P0 View		
Server: ASPSQL04\ASPSQL04	Permissions for the C	ISDOK-		G	olumn Permis:	sions
Connection: ES090NLINE\administrator	Explicit	300K.				0010110
We We connection properties	Permission	Grantor	Grant	With Grant	Deny	^
YT <u>wew connection properties</u>	References					
	Select					
	Select	dbo				
Progress	Take ownership					
All Boody	Update					
ricady	View definition					
, of he.	Law a second		_	-	_	~

#### 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 - 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 zažádat 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 vyskytují nepřípustné znaky, je třeba je převést tzv.URL encodingem. Viz např. <u>https://www.urlencoder.org/</u>.

## 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,
            "VLSUBJOBRAT": 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,
            "VLSUBJOBRAT": 0,
            "VLSUBJPOCETZAM": 0,
            "SUBJEKT WEBSITE": "www.eso9.cz",
            "VLPRAVNIFORMA": 1,
            "VLEDIPROVIDER": 1,
            "VLEDIFORMAT": 2,
            "DATOVASCHRANKAID": "czxfpqh",
            "IDOBECCIS 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: <u>https://{Server}/JsonAPI/Data/getData/{DataSource}</u>

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

**Parametr {DataSource}:** jméno tabulky nebo pohledu z ESO9, jehož data funkce vrací. Pokud používáme pro uživatelský účet přistupující k aplikační databází oprávnění pro čtení, není třeba nic dále nastavovat. Pokud používáme pro uživatelský účet přistupující k aplikační databází 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: <u>https://{Server}/JsonAPI/Data/getDataWithCondition/{DataSource}?{Condition}</u>

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

**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 getRowsCount

Vrací počet řádků tabulky nebo pohledu.

URL: https://{Server}/JsonAPI/Data/getRowsCount/{DataSource}

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

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

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

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

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: <a href="https://{Server}/JsonAPI/File/getFile/fileGUID">https://{Server}/JsonAPI/File/getFile/{fileGUID</a>

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 MultipPart část s názvem *"file"*, viz také příklad níže.

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

**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 RVNPOUV4Y2VsQVBJOkhlc2xvMTIz
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 <u>https://swaqqer.io/docs/specification/describing-request-body/multipart-requests/</u>.

#### 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 <u>https://curl.se/download.html</u>.

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/ES09JsonAPI/File/setFile?userDBProcedure=spUploadFile"
    --user ES09ExcelAPI: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.

```
$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: <u>https://api.eso.cz/ESO9JsonAPI/Data/getData/QHDOK\_FAV</u>.

### 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:

H	⊜-∂-∓									Sešit1 - Exc	el
Sout	or Domů Vložení	Rozložení stránky	Vzorce	Data	Revize	Zobraze	ní Nápo	ověda	Zátěžový test	Power Piv	vot Ç
Načís data	t Z textu/ Z Z tabulky CSV webu nebo oblas	Poslední Exist ti zdroje připo	ující Aktual ojení vše	izovat	Dotazy a pi Vlastnosti Upravit pro	<b>řipojení</b> opojení	2↓ <mark>Z A</mark> Z↓ Seřadit	Filtr	T <sub>⊷</sub> Vymazat Tœ Použít znovi T∞ Upřesnit	Text do sloupců	Dynamic doplňov
	Z <u>e</u> souboru	V Ze seš	iitu		pojení		2	seřadit a f	iltrovat		
	Z <u>d</u> atabáze	• Z <u>t</u> ext	u/CSV								
	Z <u>A</u> zure	🕨 📑 Ze sou	ıboru <u>X</u> ML		G	н	1	J	к	L	м
	Z online <u>s</u> lužeb	Z form	nátu <u>J</u> SON								
	. Z j <u>i</u> ných zdrojů	Ze slo	žky Z form	nátu JSOI	N						
	<u>K</u> ombinovat dotazy	V 🔥 Ze <u>s</u> lo	žky s JSON.	ñuje naim e souboru	portovat ve formátu						
1	Sp <u>u</u> stit Editor Power Query					_					
徵	<u>P</u> rohledat katalog dat							_			
ណា	<u>D</u> otazy z mého katalogu dat										
-	<u>N</u> astavení zdroje dat										
-	<u>M</u> ožnosti dotazu										
14											
16											

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

🗵 Importovat data					×
← → ヾ ↑ 💻 > Tento počítač			5 V	🔎 Prohledat: Tento po	pčítač
Uspořádat 🔻					
🕂 Downloads 🖈 ^ 🗸 Složk	v (7)				
🖹 Dokumenty 🖈 📰 Obrázky 🖈	3D objekty	Desktop	<b>e</b>	Dokumenty	
Affetuoso		<mark></mark>	-		
JSON API	Downloads	Hudba		Obrázky	
📙 Setkání zaměstn		<b>– – /</b>			
📙 Výměna dokladů	B Videa				
Microsoft Excel					
OneDrive - Persor	ení a jednotky (2)				
Desktop	Místní disk (C:)	Backup (D:)			
📙 Dokumenty	564 GB volných z 930 GB	1,87 TB volných z 2	2,72 TB		
📙 E-mailové příloh 🗸					
Název souboru:	https://api.eso.cz/ESO9JsonAPI/Data/c	getData/QHDOK_FAV	~	Všechny soubory (*.*)	$\sim$
			Nástroje 🔻	Importovat	Zrušit

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:

x 🛛 🕴 🙂	- <del>-</del> -						Nástroje záznamu	QHDOK_FAV – Editor Power Query		$\times$	
Soubor	Domů	Transform	nace	Přidání	í sloupce	Zobrazení	Převod				~ ?
Do tabulky Převod											
Dotazy [1]	DOK_FAV		X QHDO	VK_FAV	fx =	= Json.Document	:(Web.Contents(")	nttps://api.eso.cz/ESO9JsonAPI/Data/getData/QHDOK_FAV")) 🕔		Nastavení dotazů <b>VLASTNOSTI</b> Název QHDOK_FAV Všechny vlastnosti <b>POUŽITÝ POSTUP</b> Zdroj	*

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

x 🛛   🙂	🚺   🙂 – 🗧   QHDOK_FAV – Editor Power Query												
Soubor	Domů	Transform	ace Při	ce Přidání sloupce Zobrazení									
Zavřít a načíst •	Aktualizovat náhled 🗸	Clastn Clastn Rozšíř Sprave	iosti řený editor ovat <del>*</del>	Vybrat sloupce	Zachovat Odebrat řádky v řádky v	A Z↓ Z↓							
Zavřít		Dotaz		Správa sloupců Omezení počtu řádl									
Dotazy [1		<	X ✓ $f_{x}$ = Record.ToTable(Zdroj)										
	IDOK_FAV			lame 🔤	ABC Value	1 <b>1</b> 1							
			1	Rozbalit o	lo nových řá	ádků							
			Extrahovat hodnoty										

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:

x 🛛 🛛 🙂	I et al. 2019 and a state of the state o														
Soubor	Domů	Transform	iace Pi	řidání sloupce	Zobrazer	ní									^ <b>?</b>
Zavřít a načíst <del>•</del>	Aktualizova náhled •	Vlastn Rozšíř Sprave	nosti řený editor ovat ▼	Vybrat sloupce •	Odebrat Z	achovat Odebrat řádky v řádky v	2↓ ∡↓	Rozdělit Seskupit sloupec * podle	Datový typ: Jakékol Doužít první řác J <sub>32</sub> Nahradit hodno	i položky 🕶 lek jako záhlaví 👻 oty	Kombinovat	Spravovat parametry •	Nastavení zdroje dat	Poslední :	ij ▼ zdroje ▼
Zavřít		Dotaz		Správa s	loupců Or	nezení počtu řádk	ů Seřadit		Transformace			Parametry	Zdroje dat	Nový do	taz
Dotazy [1		<	×	✓ fx	= Table.Exp	andRecordColu	nn(#"Rozbi	alené Value", "Va	lue",			Nasta	ivení dot	azů	×
	HUOK_PAV		⊞- A <sup>B</sup> C	Name 🔄 👻	123 IDHDOK		ок 🗾	123 IDUCTOBD	123 CIS_DOK	123 IDSUBJEKT	ABC 123 RVI	B/			
			1 QHI	DOK_FAV	2	21814	55	489	ID18070001		208	✓ VLAST	NOSTI		
			2 QHI	DOK_FAV	2	21815	55	490	ID18080001		208	Nazev			
			3 QHI	DOK_FAV	2	21816	55	491	ID18090001		208	QHDC	JK_FAV		
			4 QHI	DOK_FAV	2	21817	55	492	ID18100001		208	Všechr	iy vlastnosti		
			5 QHDO	DOK_FAV	2	21818	55	493	ID18110001		208		ρομζιτή ροςτμρ		
			6 QHI	DOK_FAV	2	1835	7	499	0119020001		null	- 1002			
			7 QHI	DOK_FAV	2	21844	56	500	1019030001		208	Zd	roj		*
			8 QHI	DOK_FAV	2	21845	1	500	VZORSVUX		null	Pře	vedené na ta	bulku	
			9 QHI	DOK_FAV	2	21846	208	500	SVUX190001		null	Ko No	zbalene Value	1	×
			10 QHI	DOK_FAV	2	21847	208	500	SVUX190002		null	~ ко	zbalene value	21	¥
			11 QHI	DOK_FAV	2	21852	9	501	SV3019040001		219				
			12 QHI	DOK_FAV	2	21853	56	502	3119050001		232				
			13 QHI	DOK_FAV	2	21854	56	502	3119050002		232				
			14 QHI	DOK_FAV	2	1859	54	502	1V190001		208				

### 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.

Ŀ	5-0				Sei	šit1 - Excel				Nástroje tabulky	Nástroje dotazů	Uŋ	ych Tomáš 📧 —	
Soul	oor Dom	nů Vložen	í Rozložení str	ránky Vzorce	Data Revize	e Zobrazení N	lápověda Zát	ěžový test	Power Pivot	Návrh	Dotaz	♀ Řekněte	mi, co chcete udělat.	우 Sdilet
Názen QHD @ Z	r tabulky: OK_FAV měnit veliko Vlastnos	ist tabulky	🛃 Souhrn pomod 🔡 Odebrat duplid 🚆 Převést na obl	cí kontingenční tabul city ast Nástroje	lky Vložit Exp průřez	portovat Aktualizovat Data extern	\Xi Vlastnosti 🎼 Otevrit v prol 👾 Odpojit ní tabulky	IIIZeči □ Řá Pr	záhlavím idek souhrnů uhované řádky	První sloupec Poslední sloupe Pruhované slou Možnosti stylů tabul	✓ Tlačítko ec ek	filtru	Styly tabulky	
	•		√ fx											
	A	в		D	Е		G							
1	lame			IDUCTOBD 🔻 CI	S DOK	▼ IDSUBJEKT ▼	RVBANKA 💌	IDBANKA 🔽	RVUCET 🔽 E	BUHDOK UCET	•	RVSSYM	Dotazy a přip	- ×
2 (	HDOK_FA	V 218	14 55	5 489 ID	18070001	208	0	141	0 1	19-7487627041				
3 (	HDOK_FA	V 218	15 55	5 490 ID	18080001	208	0	141	0 1	19-7487627041			Dotazy   Pripojeni	
4 (	HDOK_FA	V 218	16 55	5 491 ID	18090001	208	0	141	0 1	L9-7487627041			Dotazů: 1	
5 0	HDOK_FA	V 218	17 55	5 492 ID	18100001	208	0	141	0 1	L9-7487627041				
6 0	HDOK_FA	V 218	18 55	5 493 ID	18110001	208	0	141	0 1	19-7487627041				
7 0	QHDOK_FA	V 218	35 7	7 499 01	19020001		0		0				Pocet nactenych rac	IKU: 2 142
8 0	HDOK_FA	V 218	44 56	5 500 10	19030001	208	0	141	0 1	19-7487627041				
9 (	QHDOK_FA	V 218	45 1	1 500 V2	ZORSVUX		0		0					
10	HDOK_FA	V 218	46 208	3 500 SV	/UX190001		0		0					
11 (	QHDOK_FA	V 218	47 208	3 500 SV	/UX190002		0		0					
12 (	HDOK_FA	V 218	52 5	9 501 SV	/3019040001	219	0	2	0 2	2509840204				
13 (	QHDOK_FA	V 218	53 56	5 502 31	19050001	232	0	4	0 1	1303998319				
14 (	HDOK_FA	V 218	54 56	5 502 31	19050002	232	0	4	0 1	1303998319				
15 (	QHDOK_FA	V 218	59 54	4 502 1\	/190001	208	0	141	0 1	19-7487627041				
16 0	HDOK_FA	V 218	63 1	1 502 V2	ZORDZA		0		0					
17 (	QHDOK_FA	V 218	64 209	9 502 D2	Z19050001		0		0					
18 0	HDOK_FA	V 218	65 113	3 502 ZE	019050001		0		0					
19 (	HDOK_FA	V 218	66 55	5 494 ID	18120001	208	0	141	0 1	19-7487627041				
20 0	HDOK_FA	V 218	67 55	5 498 ID	19010001	208	0	141	0 1	19-7487627041				
21 (	HDOK_FA	V 218	68 55	5 499 ID	19020001	208	0	141	0 1	19-7487627041				
22 (	HDOK FA	V 218	69 55	5 500 ID	19030001	208	0	141	0 1	19-7487627041				

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.