

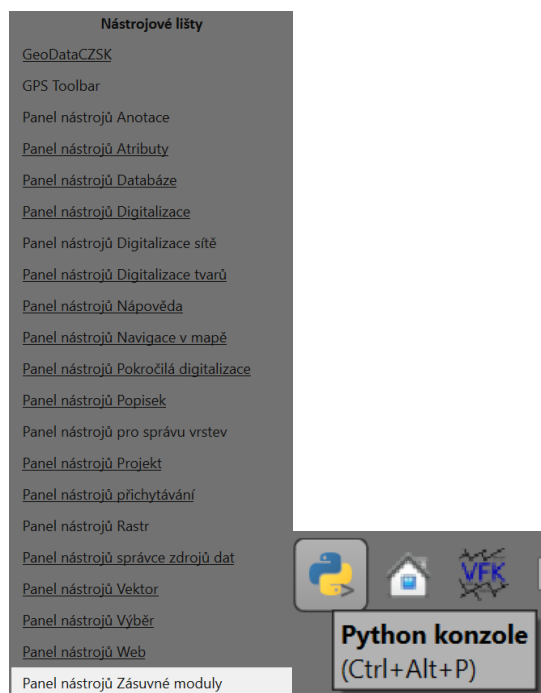
Hromadný export vrstev z GeoPackage databáze do formátu ESRI Shapefile prostřednictvím Python skriptu

Princip

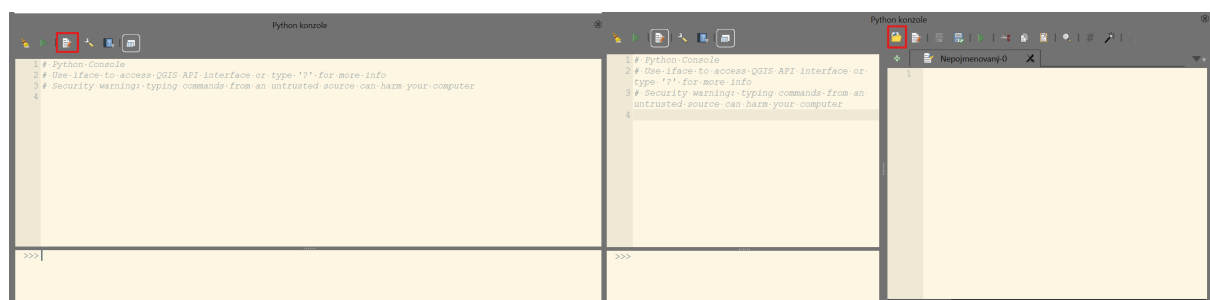
Python skript je vytvořen pro účely hromadného exportu standardizovaných a nestandardizovaných vrstev z GeoPackage databáze do jednotlivých vrstev ve formátu ESRI Shapefile, a to včetně zachování datového typu všech atributů. Výstupní vrstvy jsou exportovány s totožným pojmenováním do zvoleného cílového adresáře.

Načtení skriptu v prostředí QGIS (Python konzole)

Python skript pro hromadný export je nutné spustit přes integrovanou Python konzoli. Ujistěte se, že máte v nástrojových lištách aktivovaný panel nástrojů „Zásuvné moduly“.

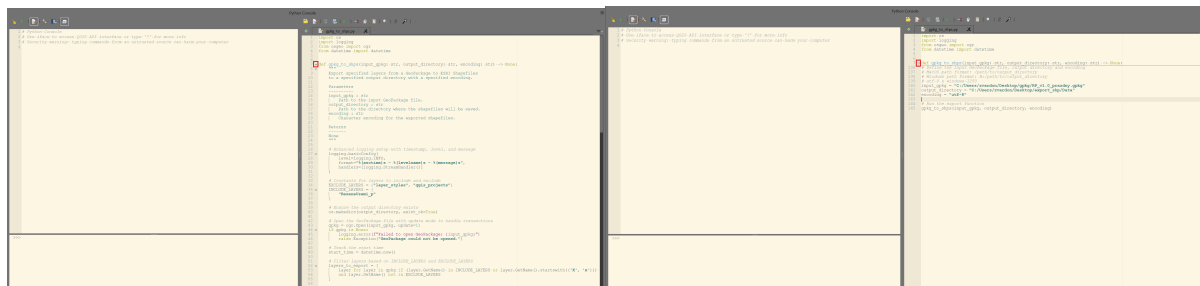


Následně Python konzoli rozšíříme o editor přes tlačítko „Zobrazit editor“ a přes editor vložíme připravený skript („gpkg_to_shps.py“).



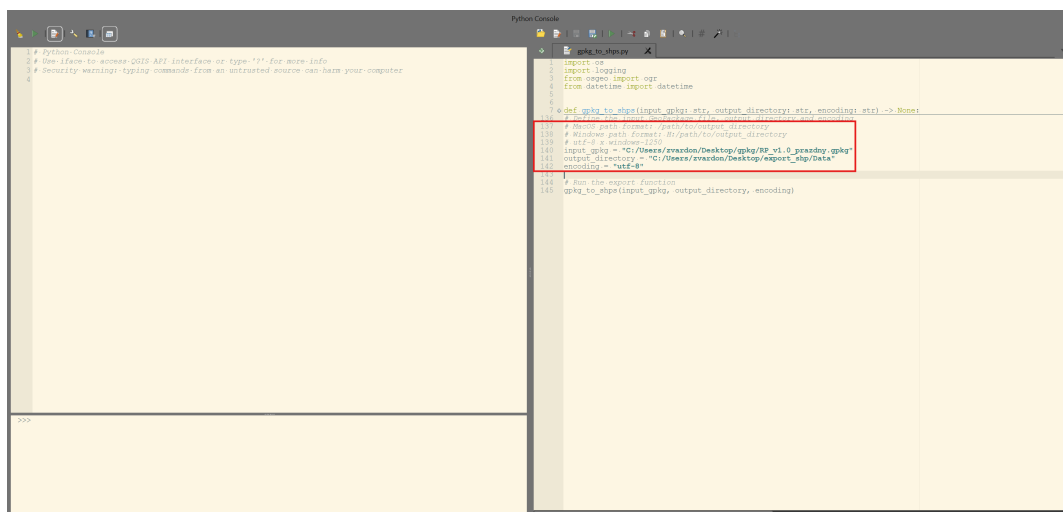
Konfigurace proměnných

Načtený skript se skládá z několikařádkového kódu, který lze, pro lepší přehled, sbalit prostřednictvím tlačítka „-“ v rámci řádku č. 7. Pro rozbalení postačí kliknout na tlačítko „+“ opět na řádku č. 7.



Pro úspěšný export vrstev z GeoPackage databáze do ESRI Shapefile je nezbytné nakonfigurovat proměnné na uvedených řádcích:

- Řádek 140: Cesta k GeoPackage souboru. Použijte předchystané formátování cesty k souboru, které se může lišit na základě používaného operačního systému (viz řádky 137 pro MacOS a 138 pro Windows).
- Řádek 141: Cesta k adresáři, kam budou výstupní vrstvy ve formátu ESRI Shapefile exportovány. Použijte předchystané formátování cesty k adresáři, které se může lišit na základě používaného operačního systému (viz řádky 137 pro MacOS a 138 pro Windows).
- Řádek 142: Kódování vrstev. Použijte předchystané formátování. Zvolte kódování dle potřeb (viz řádek 139).



Proces exportování a kontrola výstupů

Po upravení vstupních proměnných se zahájí proces exportu v rámci editoru přes ikonu „Spustit skript“. V levé části Python konzole lze sledovat průběh procesu exportu, a to jak pro jednotlivé vrstvy (viz např. řádky 7 až 9), tak i pro celý proces jako takový (viz řádky 11 až 13). Z uvedených řádků lze získat informaci o exportované vrstvě, času zahájení a ukončení exportu včetně cesty k výstupním souborům.

```
# Python Console
# One flow to access GSD API interface or type "!" for more info
# Security warnings typically come from an untrusted source - do you have your computer
# safe? (Path "C:\Users\Tom\A.14_Standards\Flow\02\pub\lady_wolf\RFP\src\env\ExportQld2\gskg_to_shps
# gw")...read_text()!
2025-04-23 13:23:11,344 ~ INFO ~ .....

# Processing layer ResourceControl_p
=====
2025-04-23 13:23:11,387 ~ INFO ~ Layer "/ResourceControl_p" successfully exported to C:/Users/random/Desktop/report_gskg_to_shps/Data/ResourceControl_p_RFP2025-04-23_13:23:11,378 ~ INFO ~ 2025-04-23 13:23:11,399 ~ INFO ~

Export process completed.
=====
2025-04-23 13:23:11,340 ~ INFO ~ Total layers processed: 2025-04-23 13:23:11,341 ~ INFO ~ Total time taken
10.650 s 2025-04-23 13:23:11,342 ~ INFO ~ Output directory: C:/Users/random/Desktop/report_gskg_to_shps/

Python Console
gskg_to_shps.py x
import os
import logging
from envs import opt
from datetime import datetime

def gskg_to_shps(input_gpkg: str, output_directory: str, encoding: str) -> None:
    """
    Export specified layers from a GeoPackage to ESBt Shapefiles
    to a specified output directory with a specified encoding.

    Parameters
    -----
    input_gpkg : str
        Path to the input GeoPackage file.
    output_directory : str
        Path to the directory where the shapefiles will be saved.
    encoding : str
        Character encoding for the exported shapefiles.

    Returns
    -----
    None
    """

    # Enhanced logging setup with timestamp, level, and message
    logging.basicConfig(
        level=logging.INFO,
        format="%asctime)s -- %i[levelname]s -- %i[message]s",
        handler=logging.StreamHandler())

    # Constants for layers to include and exclude
    INCLUDE_LAYERS = ["layer_style", "gpid_projects"]
    EXCLUDE_LAYERS = [
        "/ResourceControl_p"
    ]

    # Ensure the output directory exists
    os.makedirs(output_directory, exist_ok=True)

    # Open the GeoPackage file with update mode to handle transactions
    pkg = ogr.Open(input_gpkg, update=1)
    if pkg is None:
        logging.error(f'Failed to open GeoPackage: {input_gpkg}')
        raise RuntimeError('GeoPackage could not be opened.')

    # Track the start time
    start_time = datetime.now()
```

Vrstvy z GeoPackage databáze jsou exportovány do předem definovaného adresáře a připravené pro validaci přes kontrolní nástroj.

