Kanton St.Gallen
Baudepartement

Amt für Raumentwicklung und Geoinformation

Die Installation von pyCurl

und

der Betrieb des Python Moduls AI_uploader

zum

Upload von Interlis-Daten in die Aggregationsinfrastruktur

Dokument-Titel	Die Installation von pyCurl und der Betrieb des Python AI_uploader zum Upload von Interlis-Daten in die Aggregationsinfrastruktur
Dokument-Familie	Merkblatt
Dokument-Owner	Bernhard Ehrminger
Version	0.2
Erstelldatum	24.03.2017
Aktualisierungsdatum	05.10.2017
Abnahmedatum	
Abgenommen von	
Klassifikation	geheim / vertraulich / nicht klassifiziert
Status	in Bearbeitung / in Prüfung / abgeschlossen
Dateiname	Merkblatt_AI_Uploader.docx

Änderungskontrolle

Version	Datum	Ausführende Stelle	Bemerkungen/Art der Änderung
0.1	24.03.2017	BD-AREG /EhB	Initialversion
0.2	05.10.2017	BD-AREG / LiR	Ergänzung Parameter für POST Request: `replace_all`

Prüfung

Version	Prüfdatum	Prüfende Stelle/n	Bemerkungen

Freigabe

Version	Freigabedat.	Freigebende Stelle/n	Bemerkungen

Inhaltsverzeichnis

1 Installation von pyCurl

Zur Installation des Modules pyCurl in die mit ArcGIS 10.4 eingerichtete Python-Installation sind folgende Schritte zu unternehmen:

Herstellen einer DOS-command shell mit

```
c:\Windows\SysWOW64\cmd.exe
```

In der DOS-command shell ausführen von:

```
set PYTHONHOME=C:\Python27\ArcGIS10.4

%PYTHONHOME%\Scripts\pip install
         --index-url=http://pypi.python.org/simple/
         --trusted-host pypi.python.org pycurl
```

Die erfolgreiche Installation wird wie folgt angezeigt:

```
Collecting pycurl
  Downloading pycurl-7.43.0-cp27-none-win_amd64.whl (1.3MB)
    100% |################################| 1.3MB 3.2MB/s
Installing collected packages: pycurl
Successfully installed pycurl-7.43.0
```

2 Verwendung des Modules AI_uploader

Das Modul „AI_uploader“ besteht im Wesentlichen aus der Definition der Python-Klasse AI_uploader.
Der nachstehende Python Quellcode enthält ein vollständiges Beispiel zur Instanziierung des AI_uploaders und Aufruf seiner Methode „upload“:

```
if __name__ == '__main__':

    # Herstellen und konfigurieren eines Loggers
    logger = logging.getLogger('TEST')
    logger.setLevel('DEBUG')

    # Herstellen eines Formatters
    formater = Formatter('%(asctime)-20s | %(levelname)-10s |' +
                         ' %(name)s | %(message)s', "%d-%m-%Y %H:%M:%S")

    # Herstellen und konfigurieren eines Streamhandlers
    sh = logging.StreamHandler()
    sh.setFormatter(formater)

    # Fertigstellen des Loggers zur Ausgabe der Log-Meldungen auf der Console
    logger.addHandler(sh)

    # das hochzuladende Zip-Archiv
    zipArchiv = r'O:\Projekte\GI\Aggregationsinfrastruktur_KKGEO' + \
                os.sep + 'REST_API\Testdaten\Schaffhausen\KbS.zip'

    # Herstellen einer Instanz mit Angabe der Zielplattform
    # (Entwicklung oder Produktion) und des logger's
    uploader = AI_uploader(environment='Entwicklung', logger=logger)

    # Ausführen des Uploads
    uploader.upload(interlis_topic='kbs',
                    lv95_zip_file=zipArchiv,
                    replace_all=False,
                    publish=False)
    pass
```

Die gegenwärtige Version des AI_uploaders unterstützt nur den Upload von LV95-Daten. Eine ggf. erforderliche Erweiterung für LV03-Daten ist mit wenig Aufwand möglich. Der Betrieb des AI_uploaders mit einem Logger, der wie oben gezeigt mit „DEBUG“ konfiguriert ist, veranlasst den Aufruf von pyCurl im „verbose Mode“. Die Konfiguration des Loggers mit anderen Werten (z.B. INFO) schaltet den „verbose Mode“ von pyCurl aus.

Unsere verschlüsselten HTTPS-Verbindungen werden durch den Proxy überwacht (TLS-Terminierung), eine mit Default-Parametrisierung aktivierte Sicherheits- resp. Zertifikatüberprüfung durch pyCurl scheitert deshalb. Gegenwärtig ist daher die Prüfung durch pyCurl mit den folgenden Zeilen deaktiviert:

```
c.setopt(c.SSL_VERIFYPEER, 0)
c.setopt(c.SSL_VERIFYHOST, 0)
```

In Szenarien, in denen keine Terminierung von TLS-Verbindungen auftreten, sollten obige Programmzeilen entfernt werden können.