Rastio

Rasterdaten I/O
Schnittstellenbeschreibung

toposoft GmbH

19. Oktober 2016

Inhaltsverzeichnis

1 Einleitung

Die Raster-Zeitreihen-Datenbank TopoRast importiert Rasterdaten aus Dateien und stellt Zeitreihen zur Verfügung. Ein Export von Rastern ist nicht vorgesehen.

Rastio ermöglicht nun den Im- und Export von Rastern über eine Netzwerkschnittstelle und das Erzeugen von Rastbildern.

Rastio wird per URL gesteuert und liefert XML, PNG oder TIFF. Die Befehle sind im Folgenden berschrieben.

2 Befehle von Rastio

Befehle werden an Rastio in Form einer URL geschickt, die den Hostnamen, den Port, ggf. den Namen einer Raster-Datenbank sowie die Funktion beinhaltet. Diese URL muss mit dem http-Protokoll als GET (oder ggf. POST) versendet werden. Beispiel

```
http://rastio.server.net:8050/tt?Cmd=GETPNG&Time=1.1.2013_0:00
```

Das Rastio-Protokoll umfasst folgende Befehle:

- GET - Liest ein Raster im GeoTIFF-Format aus der Datenbank
- DIST - Liest ein Raster im GeoTIFF-Format aus der Datenbank
- PUT - Schreibt ein oder mehrere Raster im GeoTIFF-Format in die Datenbank
- GETPNG - Liest ein Raster als PNG-Bild aus der Datenbank
- DISTPNG - Liest ein Raster als PNG-Bild aus der Datenbank
- GETMULTI - Liest mehrere Raster aus der Datenbank als Multi-GeoTIFF
- FLUSH - Schließt alle Schreibevorgänge in die Datenbank ab.

Die Beispiele in den folgenden Befehlsbeschreibungen nutzen alle die Raster-ZR-Datenbank `tt` mit dem Parameter `Temperatur`.

2.1 GET

Liefert aus einer Raster-ZR-Datenbank zu einer bestimmten Zeit und ggf. einem bestimmten Parameter ein Raster im GeoTIFF-Format. Das Ergebnis wird im http-Format als image/tiff verschickt. Die Anfrage ist ein http-GET.

Parameter sind

Time der Zeitpunkt, an dem das Raster in der Datenbank abgelegt ist. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

Param Nur nötig, wenn in der Datenbank mehrere Parameter (z.B. Temperatur und Luftfeuchte) abgelegt sind. Ohne Angabe des Parameters und bei mehreren Parametern in der Datenbank wird der erste Parameter herangezogen.

Beispiel: `http://host:8050/tt?Cmd=GET&Time=17.8.2014_12:35&Param=Temperatur`

Verwendete TIFF-Tags:

0100 Anzahl Spalten 16bit unsigned

0101 Anzahl Zeilen 16bit unsigned

0102 32 bits pro Wert, 16bit unsigned

0103 1 = keine Komprimierung, 16bit unsigned

0153 3 = 32bit IEEE 754 floats, 16bit unsigned

830e Kachelbreite/Kachelhöhe/0 64bit IEE 754 double float

8482 Referenzpunkt 0/0/0/lux/loy/0 64bit IEE 754 double float

87b1 ASCII: Zeitpunkt|Parameter|Einheit|Zeitschritt

a481 ASCII: „4e37“ ist Lücke

2.2 DIST

Liefert aus einer Raster-ZR-Datenbank zu einem bestimmten Zeitintervall, einer Auswertungsart und ggf. einem bestimmten Parameter ein Raster im GeoTIFF-Format. Das Ergebnis wird im http-Format als image/tiff verschickt. Die Anfrage ist ein http-GET.

Parameter sind

From der Beginn des Bereichs, der ausgewertet werden soll. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

To das Ende des Bereichs, der ausgewertet werden soll. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

Stat die Auswertungsart: Summen (SUM), Mittelwerte (MIT), Minima (MIN) oder Maxima (MAX). Wird dieser Parameter nicht angegeben, werden Mittelwerte gebildet.

Param Nur nötig, wenn in der Datenbank mehrere Parameter (z.B. Temperatur und Luftfeuchte) abgelegt sind. Ohne Angabe des Parameters und bei mehreren Parametern in der Datenbank wird der erste Parameter herangezogen.

Beispiel:

`http://host:8050/tt?Cmd=DIST&From=27.10.2015_7:00&To=28.10.2015_7:00&Stat=Sum&Param=Niederschlag`

2.3 PUT

Schreibt ein oder mehrere Raster in die Raster-ZR-Datenbank ab einer bestimmten Zeit und ggf. zu einem bestimmten Parameter. Die Raster werden mittels http-POST binär im GeoTIFF-Format nahtlos hintereinander übermittelt. Als Ergebnis wird eine XML-Antwort geschickt, die OK enthält oder eine Fehlermeldung, z.B. *invalid grid time*.

Beim Import werden Raster, die von der Größe nicht zur Raster-ZR-Datenbank passen, abgewiesen.

Parameter der URL sind

Time der Zeitpunkt, ab dem die Raster in der Datenbank abzulegen sind. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

Param Nur nötig, wenn in der Datenbank mehrere Parameter (z.B. Temperatur und Luftfeuchte) abgelegt sind. Ohne Angabe des Parameters und bei mehreren Parametern in der Datenbank wird der erste Parameter herangezogen.

NoFlush Sinnvoll, wenn viele Raster hintereinander verschickt werden. Die Datenbank bleibt dann offen und führt also kein zeitaufwändiges Flush durch. Nach dem PUT des letzten Rasters muss dann ein abschließender FLUSH-Befehl erfolgen

Beispiel: `http://host:8050/tt?Cmd=PUT&Time=17.8.2014_12:45&Param=Temperatur` mit anschließenden Binärdaten im GeoTIFF-Format.

Für alle Raster werden die im intro der Raster-ZR-DB hinterlegten Werte benutzt. Es ist jedoch möglich, den Zeitpunkt und den Parameter pro Raster im GeoTIFF anzugeben. Dies geschieht im TIFF-Tag 87b1 (siehe Befehl GET).

2.4 GETMULTI

Liefert aus einer Raster-ZR-Datenbank zu einem bestimmten Zeitraum und ggf. einem bestimmten Parameter mehrere Raster im Geo(Multi)TIFF-Format. Das Ergebnis wird im http-Format als image/tiff verschickt. Die Anfrage ist ein http-GET.

Parameter sind

From der Zeitpunkt, an dem das erste Raster in der Datenbank abgelegt ist. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

To der Zeitpunkt, an dem das letzte Raster in der Datenbank abgelegt ist. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

Param Nur nötig, wenn in der Datenbank mehrere Parameter (z.B. Temperatur und Luftfeuchte) abgelegt sind. Ohne Angabe des Parameters und bei mehreren Parametern in der Datenbank wird der erste Parameter herangezogen.

Subst Ersatzwert für fehlende Raster. Die Angabe eines Ersatzwertes legt gleichzeitig fest, dass Raster in einem festen Zeitraster geliefert werden. Dessen Schrittweite ergibt sich aus TimeStep im Intro. Fehlen Raster, werden Ersatzraster konstruiert, die in jeder Kachel den Ersatzwert enthalten. Raster, die nicht auf glatten ZP liegen (maßgeblich ist From), werden auf den nächsten glatten ZP gerundet. Raster, die nach dieser Rundung doppelt sind, werden gelöscht.

Beispiel: `http://host:8050/tt?Cmd=GETMULTI&From=17.8.2014_12:00&To=18.8.2014_12:00`

Pro Raster werden die TIFF-Tags wie beim Befehl GET beschrieben benutzt.

2.5 FLUSH

Die Datenbankdatei wird geschlossen. Dieser Befehl ist nötig, wenn Raster geschrieben werden mit der NoFlush-Option. So wird nicht nach jedem Raster die Datei zeitaufwändig geschlossen, sondern nur abschließend einmal. Die Anfrage ist ein http-GET.

Der Befehl hat keine weiteren Parameter.

Beispiel: `http://host:8050/tt?Cmd=FLUSH`

2.6 GETPNG

Liefert aus einer Raster-ZR-Datenbank zu einer bestimmten Zeit und ggf. einem bestimmten Parameter ein Bild im PNG-Format. Die Anfrage ist ein http-GET.

Parameter sind

Time der Zeitpunkt, an dem das Raster in der Datenbank abgelegt ist. Das Format ist frei, Leerzeichen müssen durch _ ersetzt werden. Die Angabe ist obligatorisch.

Param Nur nötig, wenn in der Datenbank mehrere Parameter (z.B. Temperatur und Luftfeuchte) abgelegt sind. Ohne Angabe des Parameters und bei mehreren Parametern in der Datenbank wird der erste Parameter herangezogen.

RGB1 der Farbwert der Minimumfarbe in Hexadezimalform. Voreinstellung ist `FF0000`, also Rot.

RGB2 der Farbwert der Maximumfarbe in Hexadezimalform. Voreinstellung ist `0000FF`, also Blau.

MINVAL der Minimalwert. Voreinstellung ist $-27,5$. Werte des Rasters, die kleiner oder gleich sind, werden in RGB1 dargestellt.

MAXVAL der Maximalwert. Voreinstellung ist 100. Werte des Rasters, die größer oder gleich sind, werden in RGB2 dargestellt.

Das Ergebnis ist ein Bild im PNG-Format, dass zu jeder Kachel im Raster ein Pixel in den Farben RGB1 bis RGB2 in 255 Farbschritten enthält.

Beispiel:

`host:8050/tt?Cmd=GETPNG&Time=17.8.2014_12:45&RGB1=00FF00&RGB2=00FFFF&MINVAL=0&MAXVAL=30`

2.7 DISTPNG

Liefert aus einer Raster-ZR-Datenbank zu einem bestimmten Zeitintervall, einer Auswertungsart und ggf. einem bestimmten Parameter ein Bild im PNG-Format. Die Anfrage ist ein http-GET. Die Parameter entsprechen denen von GETPNG, es werden jedoch statt Time die Grenzen des Intervall From und To angegeben und die Art der Auswertung Stat (siehe dazu den Befehl Dist).

Beispiel:

`host:8050/tt?Cmd=DISTPNG&From=27.10.2015_7:00&To=28.10.2015_7:00`
`&Stat=Mit&RGB1=00FF00&RGB2=00FFFF&MINVAL=0&MAXVAL=30`

3 Format der Anfrage

3.1 Authentifizierung

Im HTTP-Header einer Anfrage werden in einer Zeile Benutzername und Password im Format „`Authorization: Basic username:password`“ übergeben, wobei die Zeichenkette „`username:password`“ nach Base64 (RFC 3548, Kapitel 3) kodiert wird.

Beispiel:

```
GET /tt?Cmd=FLUSH HTTP/1.0
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ
```

Von Rastio wird der Benutzer mit dem angegebenen Passwort überprüft. Sollte das Passwort falsch sein oder der Benutzer nicht existieren, wird eine „HTTP-401“-Rückmeldung generiert. In einem Browser führt dies zu einer Login-Maske.

4 XML-Format

XML-kodierte Antworten von Rastio beginnen mit einem HTTP-Header, gefolgt von den XML-kodierten Daten.

Beispiel:

```
HTTP/1.0 200 OK
Date: Tue, 28 Oct 2009 13:49:10 GMT
Server: RASTIO/1.0 (UNIX)
Last-Modified: Tue, 28 Oct 2003 13:49:10 GMT
Expires: Tue, 28 Oct 2009 13:49:10 GMT
Cache-Control: max-age=0
Connection: close
Content-Length: 6733
Content-Type: text/plain; charset=ISO-8859-1

<?XML version="1.0" encoding="ISO-8859-1"?>
<RIOR>
<OK>OK</OK>
</RIOR>
```

Zeilen sind mit einem Linefeed (ASCII-Char 10) abgeschlossen.

5 Rastio starten und einrichten

5.1 Starten und beenden

Auf dem Server wird rastio folgendermaßen gestartet:

```
rastio [option] [&]
```

Das &-Zeichen am Ende ermöglicht bei Bedarf, dass der gestartete rastio-Prozess als Dienst im Hintergrund weiterläuft.

Beim Start werden eine Reihe von Statusmeldungen ausgegeben.

Üblicherweise wird als Rastio-Port der Port 8050 genutzt.

Die Authentifizierung des Clients, der eine Abfrage an den Rastio-Server richtet, ist üblicherweise aktiviert (`Authentication on`).

Zum Beenden des im Hintergrund laufenden Rastio-Dienstes muss man den `killall`-Befehl verwenden.

```
>killall rastio
```

5.2 Optionen

Beim Start kann derRastio-Dienst anhand einiger Optionen gesteuert werden. Die folgende Übersicht wird auch ausgegeben, wenn man `rastio` mit nicht bekannten Optionen versucht zu starten:

```
usage: rastio [OPTION]
         -h, --help       : shows this list
         -v               : output version & compilation date
         -p <port>        : use port <port> rather than 8050
         -noauth          : disable authentication
         -nowrite         : disable all write access
         -verbose         : be more verbose on init
         -startdir <dir>  : starts in <dir>
```

Im Klartext:

- -h: Hilfe:
  Hier wird die Hilfe auf der Konsole ausgegeben.
- -v: Version:
  Zeigt die Version von Rastio an.
- -verbose: Verbose:
  Veranlasst Rastio, Anfragen auf der Konsole auszugeben.
- -noauth: Keine Anmeldung:
  Mit dieser Option wird die Angabe von Benutzer und Passwort nicht mehr gefordert. Siehe Kapitel 3.1.

- `-nowrite`: Lesezugriff:
  Erlaubt nur lesenden Zugriff auf die Daten.

- `-p <port>`: Port:
  Definiert den Port, auf welchem Rastio erreichbar ist. Der Standardport ist Port 8050.

- `-startdir <dir>`: Startverzeichnis:
  Startet rastio in einem anderen Verzeichnis