Universeller REST-API-Client

# REST-Client Programm

Das REST-Client Programm kann jede REST Schnittstelle sowohl vom Intranet als auch vom Internet anfragen.

## Aufbau

Die folgenden Eingabeparameter haben folgende Bedeutungen, Eingabeformate und Beispiele:

* -c
  + REST-URL, die absolut oder relativ ist. Falls die URL relativ ist, muss der Parameter -s, und wenn notwendig oder vorhanden, auch noch den Parameter -p bei der Eingabe hinzugefügt werden.
    - ######################################################
* -s
  + REST-Server, der mit dem HTTP-Protokoll angegeben werden muss.
    - http://anut240
* -r
  + REST-Config-Datei, die zum Vergleichen der REST-Bedingungen und die Ausgabe des Statuscodes dient.
    - OeNB\RestConfig\_Automic.config
* -p, --port
  + REST-Port. Die Portnummer soll ein Integerwert sein.
    - 6655
* -hd, --head
  + Header, das an der HTTP-Anfrage gesetzt wird. Das Header-Arguement soll im JSON-Format sein.
    - {‘accept’:’\*/\*’,’Content-Type’:’Content-Type’:’’application/json’}
* -du, --userdatabase
  + SQL-Server-Connection-String, der analysiert wird, um mit der Datenbank zu verbinden. Das Argument hat eine bestimmtes Format, das zu befolgen ist.
    - ######################################################
* -u
  + Userdaten bestehend aus Usernamen und Passwort. Das Argument soll im JSON-Format sein.
    - {‘user’:’\*\*\*\*\*’, ‘pass’:’\*\*\*\*\*\*\*’}
* -cf
  + Config-Datei bestehend aus den Parametern -u, -ud, -k mit deren jeweilig gesetzten Werten.
    - OeNB\Config\_Automic.config
* -bd, --body
  + Body, das an der HTTP-Anfrage gesetzt wird. Das Body-Argument soll im JSON-Format sein.
* -m
  + HTTP-Methode. Dieser Parameter kann nur anerkannte REST-API-HTTP-Methode verstehen. Standard: „GET“.
    - GET, POST, PUT, DELETE
* -lg, --log
  + Log-Dateiname, der nach Belieben gesetzt werden kann. Falls die Datei bereits existiert, werden die Logdaten zur Datei hinzugefügt. Standard: „log\_restclient.log“.
    - history.log
* -ic, --ignorecert
  + Ignoriert das SSL-Zertifikat. Der Parameter muss gesetzt werden, damit das SSL-Zertifikat nicht ignoriert wird. Standard: „True“.
* -pr, --printresponse
  + HTTP-Antwort wird explizit ausgegeben. Der Parameter muss gesetzt werden, damit es funktioniert. Standard: „False“.
* -v, --verbose
  + Verbose, um eine ausführliche Ausgabe zu bekommen. Der Parameter muss gesetzt werden, damit es funktioniert.
* -t, --test
  + Testausgabe, die für eine sehr detaillierte Ausgabe verwendet wird, um potenzielle Fehler zu finden. Der Parameter muss gesetzt werden, damit es funktioniert. Standard: „False“.
* -q
  + Query, die aus dem Parameter, Operator und Wert besteht. Die Query kann mit dem Et-Zeichen mehrere Queries kombinieren. Eine Liste wird mit dem Namen und mit den eckigen Klammern kenngezeichnet und eine Dictionary mit dem Namen und mit geschwungenen Klammern.
    - data[]/name=‘Unix1‘
* -k (Noch nicht implementiert)

## Beispiel

Die folgenden Beispiele zeigen mögliche ausführbare Eingaben:

# REST-Config Datei

Die REST-Config Datei wurde erstellt, um die Benutzereingaben und die daraus resultierenden Ergebnisse mit den REST-Bedingungen aus dieser Datei zu vergleichen, und danach den entsprechenden Integerwert auszugeben. Der ausgebende Integerwert reicht von 0 bis 3. Diese Werte haben folgende Bedeutungen:

* 0 – OK
* 1 – Fehler, Error
* 2 – http-Statuscode: 405
* 3 – http-Statuscode: 404

## Aufbau

Die Datei sollte nur aus fünf REST-Bedingungen, die jeweils nur mit Abständen getrennt sind, bestehen. Die erste Zeile muss immer mit fünf bestimmten Bezeichnungen in der gleichen Reihenfolge beschrieben werden. Die erste Zeile muss wie folgt lauten: „URL httpreturn BatchReturncode Query QueryValue“. Die restlichen Zeilen können nach Belieben in korrekte Formate gesetzt und verändert werden.

Die folgenden REST-Bedingungen haben bestimmte Bedeutungen:

* URL – der relative Pfad, der angefragt wird
* httpreturn – die HTTP-Statuscode nach der Durchführung der HTTP-Anfrage
* BatchReturncode – der auszugebene Integerwert (0-3) nach der entsprechenden Überprüfung mit den restlichen REST-Bedingungen
* Query – der abzufragende Parameter
* QueryValue – der Operator und der Parameterwert

## Beispiel

Das folgende Beispiel zeigt eine klassische Ping-Abfrage:

Das folgende Beispiel zeigt eine HTTP-Anfrage mit einer Query:

Das folgende Beispiel zeigt eine HTTP-Anfrage mit einer Wildcard-Query:

# Config Datei

Die Config Datei wurde erstellt, um wichtige Zugangsdaten abzuspeichern.

## Aufbau

Die Datei besteht nur aus den drei folgenden Parametern: u, ud, k. Diese Parameter beenden jeweils die Zeile mit einem Semikolon. Danach werden die Zugangsdaten gesetzt. Die Reihenfolge, wie diese Parameter gesetzt werden, ist unbedeutend.

Die folgenden Parameter und Zugangsdaten haben bestimmte Bedeutungen:

* u – user: Es besteht aus den Zeilen „User=“, „Passwort=“, „Key=“, die dann jeweils richtig gesetzt werden. Die Reihenfolge ist wichtig, dass erstens User, zweitens Passwort, und letztens Key gesetzt wird.
* ud – userdatabase: Es besteht aus einem SQL-Server-Connection-String, das vom Hauptprogramm analysiert wird und es dann mit der jeweiligen Datenbank verbindet.
* k – kerberos: Es wurde noch nicht programmiert.

## Beispiel

Das folgende Beispiel zeigt eine Config-Datei mit allen Parametern und Benutzerdaten: