

# README BwlUserStats

Martin Westphal, [westphal@de.ibm.com](mailto:westphal@de.ibm.com)

v1.0, 10.7.2015

Über die Blueworks Live REST API lädt diese Anwendung Informationen über die Nutzeraktivitäten für einen gegebenen Zeitraum und schreibt diese als csv-Dateien (comma-separated values) in das lokale Dateisystem. Diese Daten können dann z.B. mit Excel weiter ausgewertet werden.

## Inhalt

BwlUserStats.java, BwlUserStats.class

Java Quellcode und die übersetzte Version des Programms

commons-io-2.4.jar, wink-json4j-1.3.0.jar

benötigte Bibliotheken, Quellen:

<https://commons.apache.org/proper/commons-io/>

<http://mvnrepository.com/artifact/org.apache.wink/wink-json4j>

compile.bat, run.bat

Windows Batchdateien um das Programm zu übersetzen und auszuführen.  
Läuft nach Ersetzen von „;“ durch „:“ auch als Shell-Kommando unter Unix/Linux.

## Übersetzen

Dieser Schritt ist optional. Mit einem installiertem Java SDK und den richtigen Pfadeinstellungen kann man einfach die Batchdatei „compile.bat“ aufrufen oder das enthaltene Kommando kopieren und in einem Kommandofenster aufrufen.

```
javac -cp .;commons-io-2.4.jar;wink-json4j-1.3.0.jar BwlUserStats.java
```

Dieser Aufruf erzeugt die übersetzte Version: BwlUserStats.class

## Run

Mit einer installiertem Java Runtime und den richtigen Pfadeinstellungen kann man einfach die jeweiligen Batchdateien aufrufen oder das enthaltene Kommando kopieren und in einem Kommandofenster aufrufen. Ein einfacher Aufruf ohne zusätzliche Kommandozeilenargumente ergibt zwar eine Fehlermeldung wegen fehlender Argumente, aber auch eine Ausgabe aller Optionen, z.B.

```
java -cp .;commons-io-2.4.jar;wink-json4j-1.3.0.jar BwlUserStats
```

```
ERROR: missing command line arguments, 3 arguments required
Usage: BwlUserStats <user> <password> <account> [optional_arguments]
Optional arguments:
-h          This help message
-d <path>   Directory to store csv files, default=./userstats
-s <date>   Start date (YYYY-MM-DD), default(100 days)=2015-04-02
-e <date>   End date, default(today)=2015-07-10
-sl        Skip login data
-sc        Skip comment data
-su        Skip update data
-sv        Skip view data
```

Bevor man jedoch die Batchdateien verwendet werden können, müssen noch mithilfe eines Editors die erforderlichen Argumente wie Name, Passwort und Blueworks Live Accountname angepasst werden. Der Befehl in „run.bat“ wird dann Informationen zu Aktivitäten der letzten 100 Tage herunterladen.

Über die optionalen Befehlsargumente lässt sich das Verhalten detaillierter steuern. Mit dem Flag „-s“ kann beispielsweise das Startdatum angegeben werden. So werden mit dem folgenden Befehl Information ab dem Monatsersten (im Beispiel Juli) bis heute (Default) heruntergeladen.

```
java -cp .;commons-io-2.4.jar;wink-json4j-1.3.0.jar BwlUserStats  
YOURUSERNAME YOURPASSWORD YOURACCOUNT -s 2015-07-01
```

```
User statistics for Blueworks Live account IBM70 requested by user westphal@de.ibm.com  
Will store files in directory: ./userstats  
Period: 2015-07-01 ... 2015-07-10  
-----  
Retrieving info for 2015-07-01 ... 2015-07-10  
=> 34 login records found  
=> 2 comment records found  
=> 678 update records found  
=> 173 view records found  
-----  
Found 34 login records and stored in logins.txt  
Found 2 comment records and stored in comments.txt  
Found 678 update records and stored in updates.txt  
Found 173 view records and stored in views.txt  
DONE
```





Beachten Sie:

Da der benutzte API-Call nur einen maximalen Zeitbereich von 31 Tagen erlaubt, werden bei größeren Zeiträumen mehrere Calls für aufeinander folgende Zeiträume erzeugt. In diesen Fällen werden Sie in der Log-Ausgabe den Teil „Retrieving info for ...“ mehrfach sehen. Die Ergebnisse werden jedoch gesammelt in die Ausgabedateien geschrieben.

Wie Sie im Beispiel oben für 10 Tage sehen, können die Einträge gerade für Updates und Views für größere Zeiträume sehr zahlreich sein.

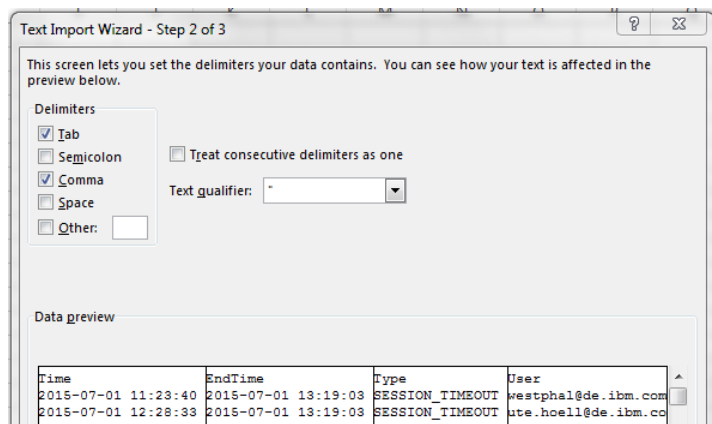
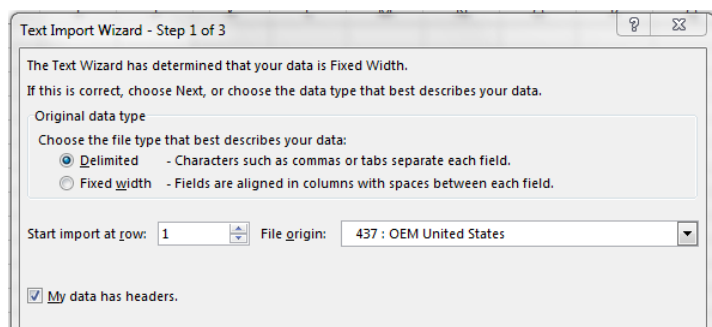
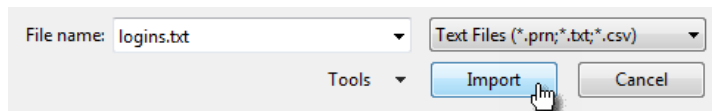
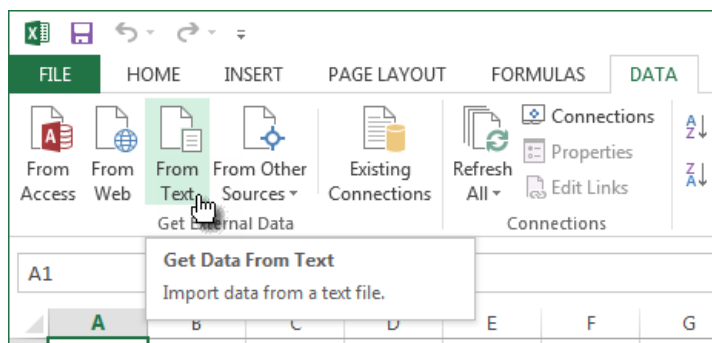
## Ausgaben

Die gefundenen Aktivitäten (oben Records genannt) werden jeweils als eine Zeile in die folgenden csv-Dateien im Ausgabeorder (Default ist ./userstats) geschrieben:

Ausgabedatei	Header
 comments.txt	Time, Space, Name, Type, Activity, User, IsReply
 logins.txt	Time, EndTime, Type, User
 updates.txt	Time, Space, Name, Type, User
 views.txt	Time, Space, Name, Type, User

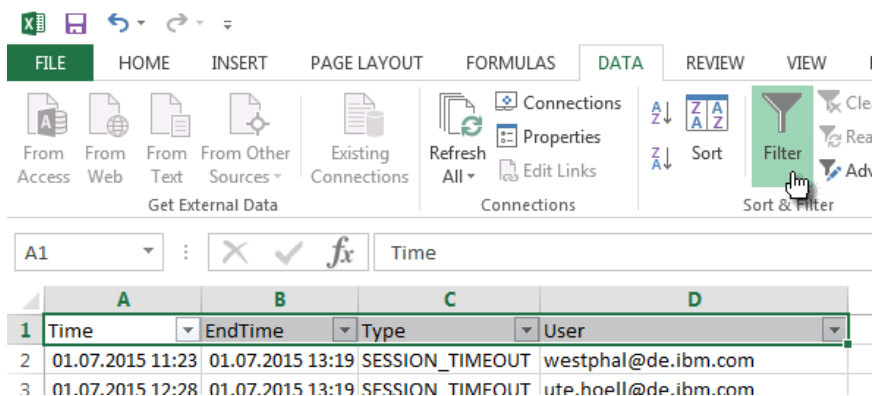
Die erste Zeile enthält jeweils einen Header mit Bezeichnungen für die durch Komma getrennten Felder. Anschließend folgen die Einträge. Die nachfolgenden Screenshots geben Hinweise wie man diese Dateien in Excel zur weiteren Analyse einlesen kann:

## Einlesen einer csv-Datei:

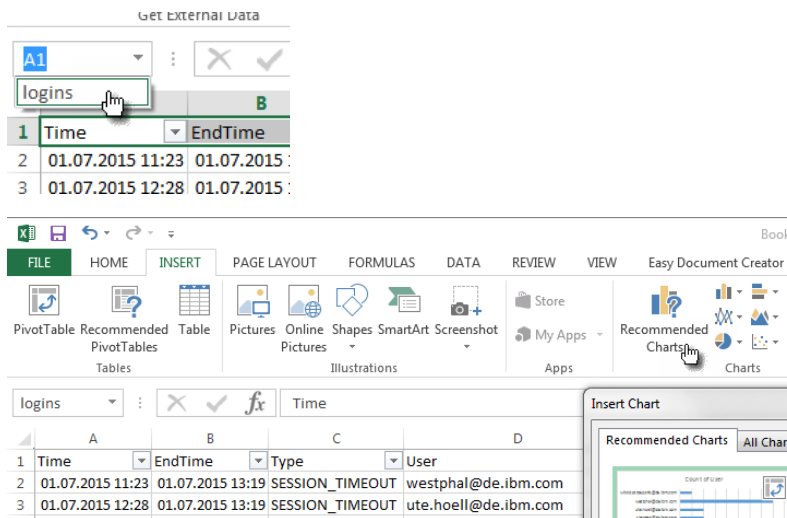


	A	B	C	D
1	Time	EndTime	Type	User
2	01.07.2015 11:23	01.07.2015 13:19	SESSION_TIMEOUT	westphal@de.ibm.com
3	01.07.2015 12:28	01.07.2015 13:19	SESSION_TIMEOUT	ute.hoell@de.ibm.com
4	01.07.2015 14:19	01.07.2015 15:44	CLIENT_TIMEOUT	barbara.reich@de.ibm.com

## Filter für den Header setzen



## Einfache Auswertung



## Refresh der Daten und Charts nachdem die Ausgabedateien neu geschrieben wurden

