Exercise 2

Using Jaql Core Operators

© Copyright IBM Corporation, 2013

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Lab 1 Using Jaql Core Operators

In this exercises you will use Jaql core operators to manipulate JSON data found in Twitter feeds.

After completing this hands-on lab, you'll be able to:

- Use the Jaql core operator to manipulate data
- Filter arrays to remove values
- Sort arrays in either ascending or descending sequence
- Write data to HDFS

Allow 20 minutes to complete this lab.

This version of the lab was designed using the InfoSphere BigInsights 2.1 Quick Start Edition, but has been tested on the 2.1.2 image. Throughout this lab you will be using the following account login information. If your passwords are different, please note the difference.

	Username	**Password**
VM image setup screen	root	password
Linux	biadmin	biadmin

1.1 Core operators

Section 1: *Core operators*

__ 1. If you have subsequently rebooted your system since you last started the BigInsights components, or you have stopped Hadoop, you will need to restart it now.

Use the icon on the desktop.

__ 2. Start the *Jaql shell.* Open a command window and execute the following:

```
cd $BIGINSIGHTS_HOME/jaql/bin
./jaqlshell
```

__ 3. Located on the lab image is a file of JSON records that contains tweets. First read this file so that you can repeatedly access the data in the file and display the contents of the file. The qualified file name is */home/labfiles/SampleData/Twitter Search.json.*

```
tweets = read(file("file:///home/labfiles/SampleData/Twitter Search.json"));
tweets;
```

__ 4. You should have received an error indicating you have an invalid character in the file name. It appears that Jaql is not too fond of spaces in file names. Code the following to get around this opportunity. Also, you did not receive the error until you displayed the data in the file. That was due to lazy evaluation. The actual read did not take place until the request to display the data.

```
tweets =
read(file("file:///home/labfiles/SampleData/Twitter%20Search.json"));
tweets;
```

__ 5. Retrieve a single field, *from_user,* using the *transform* command.

```
tweets -> transform $.from_user;
```

How did this work? When you read the file, you did not provide a schema. So how did Jaql know anything about the *from_user* field? Data in a JSON format has the field name or array name as part of the data. This allows for flexibility. If you had to code a schema for the twitter data and then later on Twitter changed the format of their data, you programs would have a problem. But since Twitter is using a JSON format, adding fields or changing the position of a field in a record has no affect.

__ 6. If you look around the beginning of each record, there is an object called *metadata.* Within that object there is a field called *result_type.* Use the *transform* operator to display the *result_type* field.

```
tweets -> transform $.metadata.result_type;
```

__ 7. Retrieve multiple fields using the *transform* operator. Now you might think that you would just list the fields that you wanted projected. That would not be correct. Multiple fields need to be encapsulated in a record or object.

```
tweets->transform {$.created_at, $.from_user, $.iso_language_code, $.text};
```

__ 8. Create a new record, *tweetsrecs* and at the same time change the name of one of the fields? Then display the contents of *tweetrecs.*

```
tweetrecs = tweets->transform {$.created_at, $.from_user, language:
$.iso_language_code, $.text};
```

```
tweetrecs;
```

__ 9. Use the *filter* operator to see all non-English language records from *tweetrecs*.

```
tweetrecs -> filter $.language != 'en';
```

__ 10. Sort *tweetrecs* so that the records appear in the order in which they were created.

```
tweetrecs -> sort by [$.created_at asc];
```

__ 11. If you notice, there are several tweets that occurred at the same time. So not only sort in created sequence but also in descending language sequence.

```
tweetrecs -> sort by [$.create_at asc, $.language desc];
```

__ 12. Aggregate your data and count the number of tweets for each language.

```
tweetrecs -> group by key = $.language into {language: key, num: count($)};
```

__ 13. You are going to write your results to a file in Hadoop but first you must create the target directory in HDFS. You can do this from the *jaqlshell* using the *hdfsShell().*

```
hdfsShell('-mkdir /user/biadmin/jaql');
```

__ 14. Write the results of the previous aggregation to a JSON file in HDFS.

```
tweetrecs->group by key = $.langauage into {language: key, num: count($)}-
>write(seq("hdfs:/user/biadmin/jaql/twittercount.seq"));
```

__ 15. Verify that your file was created.

```
hdfsShell('-ls /user/biadmin/jaql');
```

End of exercise

NOTES

NOTES

© Copyright IBM Corporation 2013.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Please Recycle