VelocityReport Plugin - Defect #652

WebService objects sort

03/06/2013 03:20 PM - Rick Bonkestoter

Status:	Rejected	**Start date:**	03/06/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Browser (if web client):			

Description

When I create a JSON object in servoy from a SQL with a sort the output on the Velocity end is different then in Servoy.

This is what I see in Servoy (application.output(context)): {Drenthe:[{name:Camping Hollandscheveld,id:0FAE4590-150C-45BC-9F91-581ED032DC3F},{name:Camping Kaspers,id:06887137-C0D4-4E06-9223-66F569CC9265}],Gelderland:[{name:Camping Lindebeek,id:BC0CDFC7-DCEF-480E-9DE2-9F1C9D272BDF},{name:Microcamping,id:59D95797-C2C5-4B39-9071-5B8A37F7A696}],Overijssel:[{name:Stoetenslagh,id:DE8F86F7-7AA4-46A3-9A51-59F22D971427},{name:Rheezerwold,id:4BD0BF28-71F6-4ACB-B4B3-28584F7ABB1B},{name:Test,id:DB628DB6-925D-4416-A7D1-265FC335104D}]}

This is what I get when I ask the json (using Velocity Webserivces): {"Overijssel":[{"id":"DE8F86F7-7AA4-46A3-9A51-59F22D971427","name":"Stoetenslagh"},{"id":"4BD0BF28-71F6-4ACB-B4B3-28584F7ABB1B","name":"Rheezerwold"},{"id":"DB628DB6-925D-4416-A7D1-265FC335104D","name":"Test"}],"Drenthe":[{"id":"0FAE4590-150C-45BC-9F91-581ED032DC3F","name":"Camping Hollandscheveld"},{"id":"06887137-C0D4-4E06-9223-66F569CC9265","name":"Camping Kaspers"}],"Gelderland":[{"id":"BC0CDFC7-DCEF-480E-9DE2-9F1C9D272BDF","name":"Camping Lindebeek"},{"id":"59D95797-C2C5-4B39-9071-5B8A37F7A696","name":"Microcamping"}]}

As you can see the object is sorted on a other way.
What can we do about this?

History

#1 - 03/06/2013 04:25 PM - Patrick Talbot

- Status changed from New to Rejected

JavaScript objects are stored in Java as HashMap, which is an unsorted collection of data: a dictionary of key/value pairs.
Try serializing with the serialize plugin and you will see an equally random unsorted result.
This is the nature of objects and I don't see what good you'll get by changing that.

I don't see an issue with that, since you usually access an object by retrieving its properties directly or by iteration.
In your case, you would either do:

```
application.output(context.Overijseel[0].id);
```

to get to the id of the Overijseel object inside the context (no matter where it is), or you would iterate using:

```
for (var name in context) {
   var objects = context[name];
   for (var i = 0; i < objects.length; i++) {
      var obj = object[i];
      application.output(obj.id);
   }
}
```

Truly in the second case, the order could be important, but it could be arbitrary when you create your object at first anyway (meaning the first time you do it, it will appear in one order, another time, it could be in another order), so you don't want to rely on properties being in a certain order, that's not safe anyway.

Arrays elements are kept in the same order though, and this is true in your example if you watch closely, look at how for the 'Drenthe' property array you always get 'Camping Hollandscheveld' followed by 'Camping Kaspers'. This will stay the same after JSON serialization and deserialization.