Servoy JasperReports Plugin - Feature #851

Use Json object as report source

05/15/2015 09:19 AM - Studio Miazzo

Status:	New	**Start date:**	05/15/2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Browser (if web client):			

Description

Since iReport 4.5.0, json is supported as a datasource (see http://jasperreports.sourceforge.net/sample.reference/jsondatasource/ and http://community.jaspersoft.com/wiki/how-query-file-based-json-datasource). However the plugin only supports JDBC connections (or foundsets, still a dataset). We'd like to see support for json added to the plugins as Servoy natively uses javascript as its language and many report tasks (i.e. calendarized views) will be made easier by using this approach rather than manipulating a dataset row by row.

As a side note, I tried to pass the report (created to use a json file datasource) a jdbc connection, ignoring it in the actual design, but the plugin won't run the report throwing instead the following exception

2015-05-14 17:46:22,034 ERROR [http-7070-2] com.servoy.j2db.util.Debug - RuntimeException in the ServoyFilter.doGet [ ]
java.lang.NoClassDefFoundError: Could not initialize class net.sf.jasperreports.engine.util.JRLoader
at com.servoy.plugins.jasperreports.JasperReportsServer.getJasperReport(JasperReportsServer.java:168)
at com.servoy.plugins.jasperreports.JasperReportsServer.getJasperPrint(JasperReportsServer.java:275)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at com.servoy.j2db.util.AbstractRemoteInvocationHandler.invokeMethod(AbstractRemoteInvocationHandler.java:77)
at com.servoy.j2db.util.ThreadingRemoteInvocationHandler$1.run(ThreadingRemoteInvocationHandler.java:79)
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
at java.lang.Thread.run(Unknown Source)