

Google™



Implementing Your Own Visualization Data Source

Itai Raz

Jesse Lorenz (Salesforce.com)

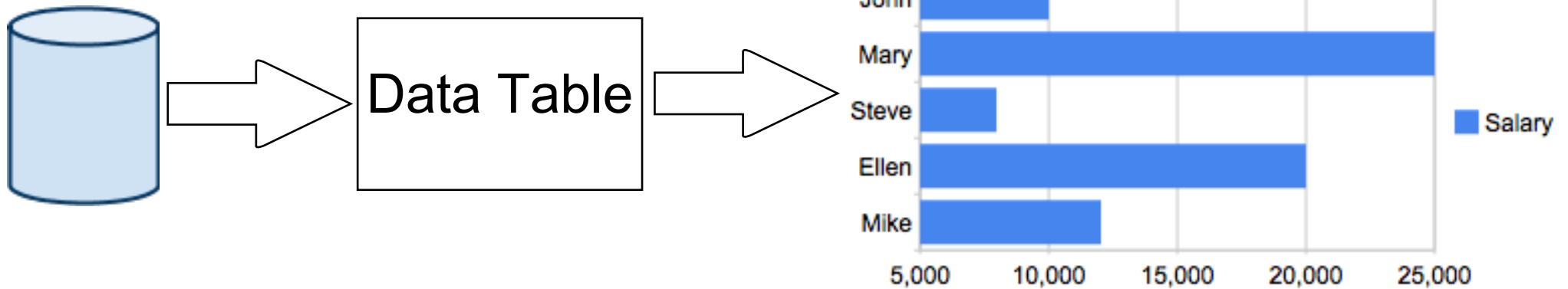
May 28, 2009



Agenda

- Introduction
- Data Source
- Java Open Source Data Source Library *new!*
- Salesforce Data Source
- Q&A

What is the Google Visualization API?



Visualizations: Many charts, many implementations, events, interactions, configurations, ...

Data table: Standard data format used by all visualizations and all data sources

Creating a DataTable (Javascript)

```
var data = new google.visualization.DataTable();  
data.addColumn('string', 'Year');  
data.addColumn('number', 'Sales');  
data.addRows(4);  
data.setValue(0, 0, '2004');  
data.setValue(0, 1, 1000);  
...  
  
var chart =  
    new google.visualization.AreaChart(div);  
  
chart.draw(data);
```

Creating a DataTable From JSON

```
var data = new google.visualization.DataTable(  
{  
  cols: [  
    {id: 'task', label: 'Task', type: 'string'},  
    {id: 'hours', label: 'Hours', type: 'number'}],  
  
  rows: [{c:[{v: 'Work'}, {v: 11}]},  
    {c:[{v: 'Eat'}, {v: 2}]},  
    {c:[{v: 'Commute'}, {v: 2, f: '2.000'}]}]  
},  
0.6 // version  
)
```

Remote Data Source

- Separate the visualization from the data
- Enable Query Language data manipulation
- Enable auto-refresh
- Using JSON constructor for improved performance
- Public protocol that deals with common issues

Sending a Query to a Data Source

```
function sendQuery() {  
    var url = 'http://viz.appspot.com/dynamic_example';  
    var query = new google.visualization.Query(url);  
    query.setQuery('select col1, col2 order by col1');  
    query.send(handleQueryResponse);  
}  
  
function handleQueryResponse(response) {  
    var data = response.getDataTable();  
    var visualization =  
        new google.visualization.Table(div);  
    visualization.draw(data);  
}
```




Remote Data Source



Wire Protocol

- The public documented format of the request and response between the visualization and the data source
- Built to support multiple versions at the same time (backward compatible)
- New version just released

New Version!

- A new version of the protocol was released now
- A new version of the Query Language too
- Both fully implemented in the java library
- Partial support also in the Python library
- All details in the docs

Security and Authentication

- It is the data source's responsibility to authenticate the user, and verify access to the data
- Options to support a few access models:
 - Script injection
 - makeRequest for gadgets (*new!*)
 - xhr to prevent XSRF (*new!*)
 - auto (*new!*)

Query Language

- The Query Language is an optional part of the protocol
- Implemented in the new java Open Source library

[http://vis.mysite.com/ds?tq=**select dept, max\(sal\) group by dept**](http://vis.mysite.com/ds?tq=select dept, max(sal) group by dept)

External Url Parameters

In many cases you may need to pass information in custom url parameters and process them in your code

<http://spreadsheet.google.com/pub?key=123&tq=select A>

- *key=123* - a custom url parameter
- *tq=select A* - a standard query languageparameter



Java Data Source Library



Java Library Concepts

- Provided as Open Source (Apache 2.0 license)
- Includes complete support of the Query Language
- Simple working examples
- Utils for SQL and CSV based data sources
- Supports both restricted (default) and non-restricted access mode
- Distributed as both source code and .jar file
- Well documented + user group



Demo time...





[Salesforce.com](https://www.salesforce.com)



For More Information

- Detailed documentation
- Many examples
- Interactive playground
- Active user group



google visualization

Google Search

I'm Feeling Lucky

[Advanced Search](#)
[Preferences](#)
[Language Tools](#)





Q & A



Google™





Appendix



Creating a Data Table (Java Library)

```
DataTable data = new DataTable() ;  
  
List<ColumnDescription> cd = new ArrayList() ;  
  
cd.add(new ColumnDescription("name", ValueType.TEXT, "Name")) ;  
cd.add(new ColumnDescription("age", ValueType.NUMBER, "Age")) ;  
  
data.addColumns (cd) ;  
  
data.addRowFromValues ("Goose", 30) ;  
data.addRowFromValues ("Duck", 10) ;  
...
```

DataSourceHelper Class

```
protected void doGet(HttpServletRequest req,
    HttpServletResponse resp) throws IOException {

    DataSourceRequest dsReq = new DataSourceRequest(req);

    DataSourceHelper.verifyAccessApproved(dsReq);
    QueryPair q = DataSourceHelper.splitQuery(dsReq.getQuery(),
        Capabilities.SELECT);
    DataTable data = createTable(q.getDataSourceQuery(), req);

    DataTable newData =
        DataSourceHelper.applyQuery(q.getCompletionQuery(),
            data,
            dsReq.getUserLocale());

    DataSourceHelper.setServletResponse(newData, dsReq, resp);
}
```


DataSourceServlet Class

```
public abstract class DataSourceServlet extends HttpServlet
    implements DataTableGenerator {

    protected void doGet(HttpServletRequest req,
        HttpServletResponse resp) throws IOException {
        DataSourceHelper.executeDataSourceServletFlow(req, resp,
            this, isRestrictedAccessMode());
    }

    protected boolean isRestrictedAccessMode() {
        return true;
    }

    public Capabilities getCapabilities() {
        return Capabilities.NONE;
    }
}
```

Query Capabilities

- Used to specify which parts of the query language you can do when creating the data table
- All others will be applied by the library
- A predefined set of options, to prevent mistakes
 - Capabilities.SQL
 - Capabilities.SELECT
 - ...
 - Capabilities.NONE
- May have significant (good) impact on latency
- May be ignored, especially by small data sources (default is NONE)

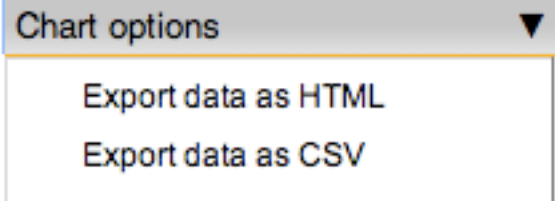
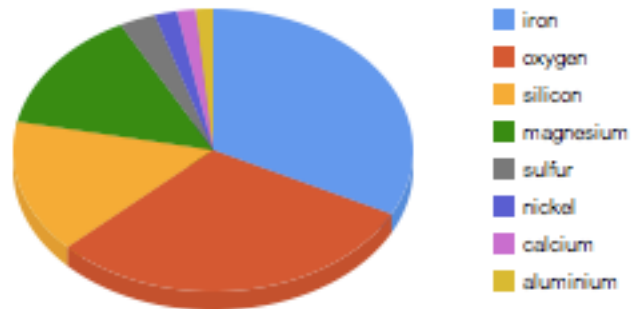
Restricted Mode

- Lets you limit the access to queries from your domain
- To be used together with cookie based authentication
- See new options in javascript to `sendQuery()`
- Data Source cannot be used from gadgets
- This is the default

Other Renderers

- In addition to output in JSON format, that is read by the javascript library, you may set the output to:
 - csv - for data export
 - html - for debug and development time
- Facilitated by the toolbar utility (in the gallery)

```
String csv = CsvRenderer.renderDataTable(dataTable, null, ',');
```



CSV and SQL Utility Classes

In addition to the simple examples, the library includes two 'real' implementation helper classes:

- *CsvDataSourceHelper* - All that is needed to use csv files as data source tables
 - Localized
 - Support for header row
- *SqlDataSourceHelper* -Helper class for MySQL database as a data source
 - Most query clauses are passed to the database
 - Can work on views