Web Client Utils - Feature #385

Passing JSON/XML FS objects to and from server

10/18/2011 04:08 PM - Bobby Drake

Status:	Closed	**Start date:**	10/18/2011
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Browser (if web client):			

Description

We need a way to pass a Servoy foundset to a web client session as XML/JSON and then a way to pass any JSON/XML object back to Servoy to update the corresponding foundset.

History

#1 - 10/18/2011 08:08 PM - P Bakker

As far as I see it, the serialization of a Foundset to JSON in order to send it to the Client and/or sending it back and merging the results back into the Foundset on the Server shouldn't be build into this plugin.

The plugin should just be extended to easily allow sending back and forth JSON and the serialization of the Foundset to JSON and/or the merging of updated coming from the browser into the Foundset should just be done in a method in the solution.

This way its way more flexible and you can control which part of the data gets send to the browser. Because most likely you don't want the whole foundset (all columns), you want some related data as well, you want display values instead of the stored integers etc.

Paul

#2 - 10/18/2011 09:58 PM - Patrick Talbot

You are right about that Paul, I'm not going to serialize a Foundset!
As I discussed with Johan about it, where would be draw the line as to what needs to be serialized? A Foundset has calculations, aggregates, and related foundset, where would be draw the line of what needs to be passed?

Actually a Dataset would be easier, but for now as you said, I will look into allowing to pass JSON back and forth, the responsability for what that JSON is is going to stay on the Servoy developer.

#3 - 10/18/2011 10:29 PM - Bobby Drake

I said "foundset", but that shouldn't be taken too literally. It would basically be a dataset. However, we do need to keep the linkage somehow between the foundet in the client's session that the dataset originated from; so that once the dataset is edited client side, we can then pass that updated JSON/XML back to the the server so it can be merged back into the foundset it originated from.

We are going to use the dataset received to fill this type of grid.
http://www.dhtmlx.com/docs/products/dhtmlxGrid/samples/14_loading_big_datasets/01_50000.html

Whichever records are edited will then be passed back to the server through JSON

#4 - 10/23/2011 02:54 AM - Patrick Talbot

- Status changed from New to Closed

Implemented in v1.3, see https://www.servoyforge.net/news/169

Example usage:

```
function onAction(event) {
    plugins.WebClientUtils.addJsReference(SERVOY_WEB_RESOURCES.JQUERY);
    var baseURL = plugins.WebClientUtils.getJSONCallbackURL(getData, {order: "company_name DESC"});
    application.output(baseURL);
    var js = '$.ajax({ \
        url: "'+baseURL+'", \
        type: "post", \
        cache: false, \
```

```
        data: JSON.stringify({id: 1, string: "hello"}), \
        contentType: "application/json", \
        dataType: "json", \
        success: function(data,textStatus,jqXHR) { \
                alert(data.columnNames) \
        } \
    });'
    plugins.WebClientUtils.executeClientSideJS(js);
}

/**
 * @param {String} jsonData the data returned by the web page
 * @param {Object} params an object map of parameters
 */
function getData(jsonData, params) {
    if (jsonData) {
        var data = plugins.serialize.fromJSON(jsonData);
        application.output(data.string);
    }
    var order = "";
    if (params) {
        var orderParam = params['order'];
        if (orderParam) {
            order = " ORDER BY "+orderParam;
        }
    }
    application.output(order);
    var ds = databaseManager.getDataSetByQuery("udm","SELECT * FROM companies"+order+";",null,-1);
    var json = plugins.serialize.toJSON(ds);
    return json;
}
```

Note that you can pass an object of parameters that will be used in the URL returned, and that the callback method will also receive parameters (extracted from the URL) and you can add your own parameters, using standard URL parameters scheme in the callback, so for example:

```
        var js = '$.ajax({ \
        url: "'+baseURL+'"+"&page=2", \
```