SolJSON

This page exists for the Solr Community to share Tips, Tricks, and Advice about the JSON Response Writer.

Reference material previously located on this page has been migrated to the Official Solr Reference Guide. If you need help, please consult the Reference Guide for the version of Solr you are using for the specific details about using this feature.

If you'd like to share information about how you use this feature, please add it to this page.
/* cwikimigrated */

JSON Response Writer

Indexing documents with JSON

See the Solr Reference Guide section JSON Formatted Index Updates.

JSON Query Response Format

JSON specific parameters

Using Solr's JSON output for AJAX

Note: also see Solr Client Libraries for higher level JavaScript clients for Solr.

Solr's JSON output makes parsing the response in JavaScript simple. Since JSON is a subset of JavaScript, one can use the built-in JavaScript parser to parse a JSON message.

```
var rsp = eval("("+jsonResponseString+")");
```

Here is a simple functional example.

To install it, place in somewhere accessible in the same server running Solr.

For example jetty server, save the text below to *webapps/ajax/ajax.html* and browse to http://localhost:8983/ajax/ajax.html

```
<html>
<head>
<title>Solr Ajax Example</title>
<script language="Javascript">
// derived from http://www.degraeve.com/reference/simple-ajax-example.php
function xmlhttpPost(strURL) {
    var xmlHttpReq = false;
    var self = this;
    if (window.XMLHttpRequest) { // Mozilla/Safari
        self.xmlHttpReq = new XMLHttpRequest();
    }
    else if (window.ActiveXObject) { // IE
        self.xmlHttpReq = new ActiveXObject("Microsoft.XMLHTTP");
    }
    self.xmlHttpReq.open('POST', strURL, true);
    self.xmlHttpReq.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
    self.xmlHttpReq.onreadystatechange = function() {
        if (self.xmlHttpReq.readyState == 4) {
            updatepage(self.xmlHttpReq.responseText);
        }
    }

    var params = getstandardargs().concat(getquerystring());
    var strData = params.join('&');
    self.xmlHttpReq.send(strData);
}

function getstandardargs() {
    var params = [
        'wt=json'
        , 'indent=on'
        , 'hl=true'
        , 'hl.fl=name,features'
        ];

    return params;
}
function getquerystring() {
  var form = document.forms['f1'];
  var query = form.query.value;
  qstr = 'q=' + escape(query);
  return qstr;
}

// this function does all the work of parsing the solr response and updating the page.
function updatepage(str){
  document.getElementById("raw").innerHTML = str;
  var rsp = eval("("+str+")"); // use eval to parse Solr's JSON response
  var html= "<br>numFound=" + rsp.response.numFound;
  var first = rsp.response.docs[0];
  html += "<br>product name="+ first.name;
  var hl=rsp.highlighting[first.id];
  if (hl.name != null) { html += "<br>name highlighted: " + hl.name[0]; }
  if (hl.features != null) { html += "<br>features highligted: " + hl.features[0]; }
  document.getElementById("result").innerHTML = html;
}
</script>
</head>
<body>

<form name="f1" onsubmit='xmlhttpPost("/solr/select"); return false;'>
  <p>query: <input name="query" type="text">
  <input value="Go" type="submit"></p>

<div id="result"></div>
<p/><pre>Raw JSON String: <div id="raw"></div></pre>
</form>
</body>
</html>
```

⚠️ **Potential UTF-8 encoding issue**

This is a small note: In some environment, the default javascript *escape()* may fail to encoded non-ASCII character into the utf-8 encoded string, which will make the search failed.

To solve the problem universally, please refer the utf-8 encoding package by webtoolkit.info to write your own encoding function: http://www.webtoolkit.info/javascript-url-decode-encode.html

*For Javascript 1.5, you can use *encodeURIComponent()* insteads of *escape()*: http://www.dangrossman.info/2007/05/25/handling-utf-8-in-javascript-php-and-non-utf8-databases/

To apply the correct encode function, just replace the *escape(query)* in function *getquerystring()*

```
qstr = 'q=' + escape(query);
```

into your own escape function (e.g. my_escape())

```
qstr = 'q=' + my_escape(query);
```

More about javascript escape function: http://xkr.us/articles/javascript/encode-compare/

CategoryQueryResponseWriter