

Code to Help JumpStart Client Server Communication

HTML JS Code with jQuery

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8"/>
5
6 <style>
7 body { background-color: yellow }
8 </style>
9
10
11 <!-- HTML Comment: Load the jquery JavaScript library -->
12 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
13
14 <script>
15
16 function callAjax(parm1, parm2) {
17     // set up URL for AJAX call
18     url = 'https://s2.smu.edu/~coyle/python/jsonJumpStart/sendJson.py';
19
20     // log for debug
21     console.log("In callAjax: parms = " + parm1 + "," + parm2);
22
23     // Make the AJAX CALL
24     // Note that we are passing a JS Object to the jQuery function $.ajax( )
25     $.ajax({
26         type: 'GET',
27         url: url,
28
29         // tell Ajax to expect json coming in from server
30         dataType: "json",
31
32         // wait for server to respond after making request
33         async: false,
34
35         // this will append ?p1=<val>&p2=<val> to URL
36         data: {p1:parm1,p2:parm2},
37
38         success: function(response) {
39             // msg will be a JS (JSON) Object with fields
40             console.log("Success:" + JSON.stringify(response));
41
42             // Update web page with data from server
43             document.getElementById("responseKey").innerHTML =
44             "The JSON returned from the server included a property test = " +
45                 response.test ;
46             document.getElementById("responseKey").style.color = "red";
47
48             document.getElementById("responseString").innerHTML = "The Response String was: "
49                 + JSON.stringify(response);
50             document.getElementById("responseString").style.color = "green";
51
52         },
53
54         error: function(jqXHR, exceptionStr) {
55             // jqXHR is a JS object with properties - of interest to us is jqXHR.status (404, 500, etc.)
56             // exceptionStr will have additional clues
57
58             if (jqXHR.status === 0) {
59                 alert('Unable to connect.\n Verify Network.');
```

```

70         'Tip: Use curl to test your server code'));
71     } else if (exceptionStr === 'timeout') {
72         alert('Time out error.');
```

73 } else if (exceptionStr === 'abort') {

74 alert('Ajax request aborted.');

75 } else {

76 alert('An Error Occurred. see below.\n' + jqXHR.responseText);

77 }

78 }

79

80 });

81 }

82

83 function talkToServer() {

84 // pull value out of TextBox1 field

85 var s1 = document.getElementById('textBox1').value;

86 var s2 = document.getElementById('textBox2').value;

87

88 console.log("talkToServer: s1=" + s1 + " s2=" + s2);

89

90 // pass parms to function that actually makes the Ajax call

91 callAjax(s1,s2);

92 }

93

94

95

96

97 </script>

98 </head>

99

100 <body>

101 <h2>Rollo Zwilling</h2>

102 <h3>12345678</h3>

103

104 Parm1: <input name="textBox1" type="text" maxlength="12" id="textBox1" class="searchField"/></p>

105 Parm2: <input name="textBox2" type="text" maxlength="12" id="textBox2" class="searchField"/></p>

106

107 <button onclick="talkToServer()">Make AJAX Call</button>

108

109 <h3 id="responseKey"></h3>

110 <h4 id="responseString"></h3>

111

112

113 </body>

114 </html>

115

Python Server Code

```

116 #!/usr/bin/env python
117
118 #IMPORT libraries
119 import cgi
120 import json
121 import sys
122 import cgi
123 import re
124
125
126 cgibt.enable()
127
128 #Tell Client what to expect either text/json or text/html
129 print "Content-Type: text/json\n"
130
131
132 #create a Python dictionary to send as JSON
133 mydict = {"test": True}
134
135 #extract paramters (keys and values) from client
136 parmObj = cgi.FieldStorage()
137
138 #iterate over keys - add key-value to mydict
139 for key in parmObj.keys():
140     mydict[key] = parmObj.getvalue(key)
141
142
143 # convert dict to json string
144 json_data = json.dumps(mydict)
145
146 # send the JSON back to the client
147 print json_data
```