

```

1  var Web3 = require('web3');
2  var web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
3  var regAddr = "0xddf27a729d05be6f11be50b1905daa6e7b508c91";
4  var regAbi = [
5      {
6          "constant": true,
7          "inputs": [
8              {
9                  "name": "_str",
10                 "type": "string"
11             }
12         ],
13         "name": "stringToBytes32",
14         "outputs": [
15             {
16                 "name": "",
17                 "type": "bytes32"
18             }
19         ],
20         "payable": false,
21         "stateMutability": "view",
22         "type": "function"
23     },
24     {
25         "constant": true,
26         "inputs": [
27             {
28                 "name": "",
29                 "type": "bytes32"
30             }
31         ],
32         "name": "lookupTable",
33         "outputs": [
34             {
35                 "name": "scName",
36                 "type": "string"
37             },
38             {
39                 "name": "subject",
40                 "type": "address"
41             },
42             {
43                 "name": "object",
44                 "type": "address"
45             },
46             {
47                 "name": "creator",
48                 "type": "address"
49             },
50             {
51                 "name": "scAddress",
52                 "type": "address"
53             },
54             {
55                 "name": "abi",
56                 "type": "bytes"
57             }
58         ],
59         "payable": false,
60         "stateMutability": "view",
61         "type": "function"
62     },
63     {
64         "constant": true,
65         "inputs": [
66             {

```

```

67         "name": "_methodName",
68         "type": "string"
69     }
70 ],
71 "name": "getContractAddr",
72 "outputs": [
73     {
74         "name": "_scAddress",
75         "type": "address"
76     }
77 ],
78 "payable": false,
79 "stateMutability": "view",
80 "type": "function"
81 },
82 {
83     "constant": true,
84     "inputs": [
85         {
86             "name": "_methodName",
87             "type": "string"
88         }
89     ],
90     "name": "getContractAbi",
91     "outputs": [
92         {
93             "name": "_abi",
94             "type": "bytes"
95         }
96     ],
97     "payable": false,
98     "stateMutability": "view",
99     "type": "function"
100 },
101 {
102     "constant": false,
103     "inputs": [
104         {
105             "name": "_methodName",
106             "type": "string"
107         },
108         {
109             "name": "_abi",
110             "type": "bytes"
111         }
112     ],
113     "name": "methodAbiUpdate",
114     "outputs": [],
115     "payable": false,
116     "stateMutability": "nonpayable",
117     "type": "function"
118 },
119 {
120     "constant": false,
121     "inputs": [
122         {
123             "name": "_methodName",
124             "type": "string"
125         },
126         {
127             "name": "_scAddress",
128             "type": "address"
129         }
130     ],
131     "name": "methodAcAddressUpdate",
132     "outputs": [],

```

```

133     "payable": false,
134     "stateMutability": "nonpayable",
135     "type": "function"
136 },
137 {
138     "constant": false,
139     "inputs": [
140         {
141             "name": "_name",
142             "type": "string"
143         }
144     ],
145     "name": "methodDelete",
146     "outputs": [],
147     "payable": false,
148     "stateMutability": "nonpayable",
149     "type": "function"
150 },
151 {
152     "constant": false,
153     "inputs": [
154         {
155             "name": "_oldName",
156             "type": "string"
157         },
158         {
159             "name": "_newName",
160             "type": "string"
161         }
162     ],
163     "name": "methodNameUpdate",
164     "outputs": [],
165     "payable": false,
166     "stateMutability": "nonpayable",
167     "type": "function"
168 },
169 {
170     "constant": false,
171     "inputs": [
172         {
173             "name": "_methodName",
174             "type": "string"
175         },
176         {
177             "name": "_sname",
178             "type": "string"
179         },
180         {
181             "name": "_subject",
182             "type": "address"
183         },
184         {
185             "name": "_object",
186             "type": "address"
187         },
188         {
189             "name": "_creator",
190             "type": "address"
191         },
192         {
193             "name": "_scAddress",
194             "type": "address"
195         },
196         {
197             "name": "_abi",
198             "type": "bytes"

```

```

199         }
200     ],
201     "name": "methodRegister",
202     "outputs": [],
203     "payable": false,
204     "stateMutability": "nonpayable",
205     "type": "function"
206 },
207 {
208     "constant": false,
209     "inputs": [
210         {
211             "name": "_methodName",
212             "type": "string"
213         },
214         {
215             "name": "_scName",
216             "type": "string"
217         }
218     ],
219     "name": "methodScNameUpdate",
220     "outputs": [],
221     "payable": false,
222     "stateMutability": "nonpayable",
223     "type": "function"
224 }
225 ]
226 var methodName = "Method1";
227 var register = web3.eth.contract(regAbi).at(regAddr);
228 var accAddr = register.getContractAddr(methodName);
229 var accAbiBytes = register.getContractAbi(methodName);
230 var accAbi = JSON.parse(web3.toAscii(accAbiBytes));
231 var myACC = web3.eth.contract(accAbi).at(accAddr);
232 var myEvent = myACC.ReturnAccessResult({fromBlock: 0, toBlock: 'latest'});
233 myEvent.watch(function(err, result){
234     if(!err){
235         //console.log(result);
236         console.log("Contract: "+result.address);
237         console.log("Block Number: " + result.blockNumber);
238         console.log("Tx Hash: " + result.transactionHash);
239         console.log("Block Hash: "+ result.blockHash);
240         console.log("Subject: "+ result.args._from);
241         console.log("Time: " + result.args._time.toNumber());
242         console.log("Message: " + result.args._errmsg);
243         console.log("Result: " + result.args._result);
244         if (result.args._penalty > 0)
245             console.log("Requests are blocked for " + result.args._penalty + " minutes!");
246         console.log("\n");
247     }
248 })
249

```