

Make a new Web by fetch method through the collection use the same url to GET/POST - version 11.6

Step 1:

Make a new function files

webui_html5-2.398.6-src/data/app/

1. Templates -> new function.html
2. Views -> new function.js
3. Collection -> new function.js
4. Models -> new function.js
5. strings/nls -> new function.js
6. root -> new function.js
7. lang_support/zh-cn -> new function.js
8. lang_support/zh-tw -> new function.js

spx_restservice-2.372.0-src/data/

1. new function.c

Step 2:

Webui_html5-2.398.6-src/data/app/ router.js

In router.js file add the new function file path

Note: that the order of joining should be the same

Step 3:

Add the new function icon in you want to display page

Main files code content

Models

```
webui_html5-2.398.6-src > data > app > models > JS fan.js > define() callback
1  define(["jquery", "underscore", "backbone", "app", "i18n!strings/nls/fan"],
2
3  function($, _, Backbone, app, locale) {
4
5      var model = Backbone.Model.extend({
6
7          defaults: {},
8
9          validation: {
10             fan1: {
11                 fn: function(value, attr, computedState) {
12                     if (value > 100) {
13                         alert(locale.invalid_value);
14                         return true;
15                     } else {
16                         return false;
17                     }
18                 },
19             },
20         },
21     });
22
23     return model;
24
25 }
```

You can set the input limit for each item in this way

Collection

```
webui_html5-2.398.6-src > data > app > collection > JS fan.js > ...
1  define(["jquery", "underscore", "backbone", "models//fan"],
2
3  function($, _, Backbone, FanModel) {
4
5      var collection = Backbone.Collection.extend({
6
7          url: function() {
8              return "/api/settings/fan"
9          },
10
11         model: FanModel
12     });
13
14     return new collection();
15
16
17 }
```

Put the url in this way

View

```
webui_html5-2.398.6-src > data > app > views > settings > JS fan.js > define() callback > view > save
1  define(['jquery', 'underscore', 'backbone', 'app',
2    //data
3    'collection//fan',
4    //localize
5    'i18n!strings/nls/fan',
6    //template
7    'text!templates/settings/fan.html',
8    //plugins
9    'iCheck', 'tooltip', 'alert'],
10
11  function($, _, Backbone, app, fanCollection, locale, fanTemplate) {
12
13    var view = Backbone.View.extend({
14
15      template: _.template(fanTemplate),
16
17      initialize: function() {
18        this.collection = fanCollection;
19        this.collection.bind('add', this.add, this);
20      },
21
22      events: {
23        "click #save": "save"
24      },
25
26      beforeRender: function() {},
27
28      afterRender: function() {
29        this.collection.fetch(); // type: GET , GET data from spx
30        this.collection.each(function(model) {
31          this.add(model, this.collection);
32        }, this);
33      },
34
35      add: function(model, collection, options) {
36        $('#idfan1').attr('value', model.get('fan1')); // GET data from model to show
37      },
38
39      save: function() {
40
41        $("#save-icon").removeClass().addClass("ion-load-d");
```

```
38
39     save: function() {
40
41         $("#save-icon").removeClass().addClass("ion-load-d");
42         this.$(".alert-success,.alert-danger").addClass("hide");
43
44         var context = this;
45         try {
46             context = that || this;
47         } catch (e) {
48             context = this;
49         }
50
51         context.collection.create({ // type: POST
52             'fan1': parseInt($('#idfan1').val(), 10), // POST data to spx
53         }, {
54             success: function() {
55                 context.$("#save-icon").removeClass().addClass("fa fa-save");
56                 context.$(".alert-success").removeClass("hide");
57                 setTimeout( function() {
58                     context.$(".alert-success").addClass("hide");
59                 },5000);
60             },
61
62             error: function() {
63                 context.$("#save-icon").removeClass().addClass("fa fa-save");
64                 context.$(".alert-danger").removeClass("hide");
65                 setTimeout( function() {
66                     context.$(".alert-danger").addClass("hide");
67                 },5000);
68             }
69         });
70     },
71     serialize: function() {
72         return {
73             locale: locale
74         };
75     }
76 });
77 return view;
78 });
```

SPX settings_fan.c

GET

```
spx_restrservice-2.372.0-src> data > C settings_fan.c> ...
1  #include "rest.h"
2
3  #define NETFN_OEM_ASRR                0x3A
4  #define CMD_ASRR_SET_SMART_FAN        0x01
5  #define CMD_ASRR_GET_SMART_FAN        0x02
6
7
8  START_AUTHORIZED_COLLECTION (getfan, GET, "/settings/fan", 1, matches, true)  // type: GET
9  {
10     uint8 pRes[9];
11     uint32 pResLen = 9;
12
13     COLLECTION_NEW_MODEL();
14
15     pRes[0] = LIBIPMI_Send_RAW_IPMI2_0_Command(&IPMISession, PAYLOAD_TYPE_IPMI,
16         ((NETFN_OEM_ASRR << 2)), CMD_ASRR_GET_SMART_FAN, NULL, 0, pRes,
17         &pResLen, DEFAULT_IPMITIMEOUT);
18
19     if(pRes[0] != CC_NORMAL) {
20         THROW_MODEL_ERROR(STATUS_500, "Error gettings fan configuration", OPERATION_CLEAR_SEL_FAILED);
21     }
22     else {
23         MODEL_ADD_INTEGER("fan1", pRes[1]);
24     }
25
26     if(0) { matches = matches; }
27
28     COLLECTION_ADD_MODEL();
29     COLLECTION_OUTPUT();
30
31 }END_AUTHORIZED_COLLECTION
32
```

POST

```
32
33 START_AUTHORIZED_MODEL (setfan, POST, "/settings/fan", 4, matches, true)  // type: POST
34 {
35     uint8 pReq[8]={0,0,0,0,0,0,0,0};
36     uint8 pRes[1];
37     uint32 dwReqDataLen;
38     uint32 dwResDataLen;
39
40     dwReqDataLen = 8;
41     dwResDataLen = sizeof(INT8);
42
43     pReq[0] = REQUEST_VAR_INTEGER("fan1");
44
45     pRes[0] = LIBIPMI_Send_RAW_IPMI2_0_Command(&IPMISession, PAYLOAD_TYPE_IPMI,
46         ((NETFN_OEM_ASRR << 2)), CMD_ASRR_SET_SMART_FAN, pReq, dwReqDataLen,
47         pRes, &dwResDataLen, DEFAULT_IPMITIMEOUT);
48
49     if(pRes[0] != CC_NORMAL) {
50         THROW_MODEL_ERROR(STATUS_500, "Error setting fan configuration", OPERATION_CLEAR_SEL_FAILED);
51     }
52
53     if(0) { matches = matches; }
54
55     SAVE_SUCCEEDED_OUTPUT();
56
57 }END_AUTHORIZED_MODEL
58
59 int settings_fan()
60 {
61     add_handler(getfan);
62     add_handler(setfan);
63     return 0;
64 }
```