



 Repositories

Create

 owner/repository ? 

 **SWEN30007 Group 4**  bairrd  Share

Clone

Branch

Pull request

...

Overview

Source

Commits

Branches

Pull requests

Issues

Wiki

Downloads

⚙

#4 **MERGED**  operatordev →  master


Unapprove 1


Merge in operator interface code to main branch

Overview

Commits

Activity

Author  Ben Chong


Reviewers 

Description Merge in operator interface code to main branch
None of this has touched the app code, so I expect this to be fine.

Stop watching

Learn about pull requests

Comments (0)



What do you want to say?

Files changed (8)

+36	-15	parseCloudCode/cloud/app.js
+16	-3	parseCloudCode/cloud/json_query_tester.py
+183	-1	parseCloudCode/cloud/main.js
+68	-0	parseCloudCode/cloud/views/alert.ejs
+4	-2	parseCloudCode/cloud/views/hello.ejs
+5	-1	parseCloudCode/cloud/views/login.ejs
+7	-6	parseCloudCode/cloud/views/system.ejs
+23	-1	parseCloudCode/public/css/style.css

parseCloudCode/cloud/app.js

Side-by-side diff View file

...

65 65

66 66 app.get('/', function(req, res) {

67 67 if (Parse.User.current()) {

68 - // No need to fetch the current user for querying Note objects.

69 - var Note = Parse.Object.extend("Note");

70 - var query = new Parse.Query(Note);

71 - query.find().then(function(results)

72 - {

73 - // Render the notes that the current user is allowed to see.

74 - },

75 - function(error) {

76 - // Render error page.

77 - });

68 + // user is logged in.

69 + res.redirect('/system');

70 +

78 71 } else {

79 72 // Render a public welcome page, with a link to the '/login' endpoint.

80 73

...

90 83 //result2 is an array of json objects.

91 84 var result = "";

92 85 var stringed = "";

93 - console.log(result2)

86 + //console.log(result2)

94 87 //for (var i=0; i<result2.length; i++)

95 88 //{

// console.log(result2.name);

```

113 106    });
114 107
115 108    app.get('/system/alert/:alarmID', function(req, res){
116 108        console.log(req.params.alarmID);
117 108        //we've got an alarmID, let's bring up the alert info along with the
118 108        res.redirect('/system');
119 108    });
120 109 +    Parse.Cloud.run('get_user_from_alert', {alertID: req.params.alarmID}, {
121 109 +        success:function(result2) {
122 109 +            //console.log(req.params.alarmID);
123 109 +            //console.log(result2);
124 109 +            //we've got an alarmID, let's bring up the user info along with it
125 109 +            var thisurl = "/system/alert/"+req.params.alarmID;
126 109 +            var result = "";
127 109 +            result = String(result2);
128 109 +            res.render('alert.ejs', {message: result2, url:thisurl});
129 109 +        },
130 109 +        error: function(error){
131 109 +        }
132 109 +    });
133 109 + });
134 109 +
135 109 + app.get('/system/alert/:alarmID/dismiss', function(req, res){
136 109 +     Parse.Cloud.run('dismiss_alarm', {alertID: req.params.alarmID}, {
137 109 +         success:function(result2) {
138 109 +             //console.log(req.params.alarmID);
139 109 +             //console.log(result2);
140 109 +             //we've got an alarmID, let's bring up the user info along with it
141 109 +             res.redirect('/system');
142 109 +         },
143 109 +         error: function(error){
144 109 +         }
145 109 +     });
146 109 + });
147 109 +
148 109 + // You could have a "Profile" link on your website pointing to this.

```

parseCloudCode/cloud/json_query_tester.py

Side-by-side diff

View file

```

9 9    print result ''
10 10
11 11
12 11 -import json,httplib
12 11 +'''import json,httplib
13 13 connection = httplib.HTTPSConnection('api.parse.com', 443)
14 14 connection.connect()
15 15 connection.request('POST', '/1/functions/get_active_alarms', json.dumps({

```

```

19 19         "Content-Type": "application/json"
20 20     })
21 21     result = json.loads(connection.getresponse().read())
22 21 -print result
22 21 +print result ''
23 23
24 24     # import json,httplib
25 25     # connection = httplib.HTTPSConnection('api.parse.com', 443)

```

```

31 31     #         "Content-Type": "application/json"
32 32     #     })
33 33     # result = json.loads(connection.getresponse().read())
34 33 -# print result
34 33 +# print result
35 33 +
36 33 +import json,httplib
37 33 +connection = httplib.HTTPSConnection('api.parse.com', 443)
38 33 +connection.connect()
39 33 +connection.request('POST', '/1/functions/get_user_from_alert', json.dumps({

```

```

40 +   "alertID": "HjHE1bsEzN"
41 + }, {
42 +   "X-Parse-Application-Id": "MVq1PZGssKpeTZAWjqvgljViUY1FE1WtyZiuVDTa",
43 +   "X-Parse-REST-API-Key": "jSnTsnBTx89Bz68dJlRy8EAFRYwpdaeNgigjGZZQ",
44 +   "Content-Type": "application/json"
45 + })
46 +result = json.loads(connection.getResponse().read())
47 +print result

```

parseCloudCode/cloud/main.js

Side-by-side diff

View file

```

35 35
36 36   require('cloud/app.js');
37 37
38 +Parse.Cloud.job("check_last_check_in_time", function(request, status) {
39 +   Parse.Cloud.useMasterKey();
40 +   var userquery = new Parse.Query("User");
41 +   var checkinquery;
42 +   userquery.equalTo("Type", "Caree");
43 +   var timeout = 60;
44 +   //var timeout = 60*60*24//in seconds - seconds, minutes, hours - 24 hours
45 +   userquery.each(function(user) {
46 +       //iterate through every user
47 +       checkinquery = new Parse.Query("Check_In");
48 +       var checkinID = user.get("lastCheckInId");
49 +       //console.log(user);
50 +
51 +       checkinquery.get(checkinID, {
52 +           success: function(checkin_object)
53 +           {
54 +               console.log(checkinID);
55 +               console.log(checkin_object);
56 +               var time_sent = new Date();
57 +               time_sent = checkin_object.updatedAt;
58 +               var diff = Math.abs(new Date() - time_sent);
59 +               diff = diff/1000; //time in seconds since last check in
60 +               console.log(diff)
61 +
62 +               if(diff>timeout)
63 +               {
64 +                   //aka, this person is over the check in timer. grab the list of active alarms to see if
65 +                   Parse.Cloud.run('get_active_alarms', {}, {
66 +                       success: function(result2) {
67 +                           //console.log(result2);
68 +
69 +                           //i have no fucking idea why this is necessary but it doesn't work otherwise
70 +                           var result3 = ""
71 +                           result3 = String(result2);
72 +                           var result = JSON.parse(result3);
73 +                           var found = -1;
74 +                           //console.log(result);
75 +
76 +                           for(var i=0; i<result.length; i++)
77 +                           {
78 +                               if(result[i].username == user.get("username"))
79 +                               {
80 +                                   console.log(result[i].username);
81 +                                   console.log(user.get("username"));
82 +                                   //ie we already have an activated alarm
83 +                                   found = i;
84 +                                   console.log('activated alarm found already');
85 +                                   break;
86 +                               }
87 +                           }
88 +
89 +                           if(found == -1)
90 +                           {
91 +                               //add new alarm object
92 +                               console.log('must add alarm!');
93 +
94 +                               var firstName = user.get("firstName");
95 +                               var lastName = user.get("lastName");
96 +
97 +                               // Send Alarm to server
98 +                               var Stuff = Parse.Object.extend("Alarm");
99 +                               var obj = new Stuff();
100 +                               obj.set("Name", firstName + " " + lastName);
101 +                               obj.set("username", user.get("username"));

```

```

102 +         obj.set("Activated", true);
103 +         obj.set("Type", "No response from patient");
104 +         obj.save(null,
105 +         {
106 +             success: function(obj)
107 +             {
108 +                 response.success("yayifications!");
109 +
110 +                 Parse.Push.send({
111 +                     channels: [user.get("username")],
112 +                     data: {
113 +                         alert: firstName + " " + lastName + " needs HEL
114 +                     }
115 +                 }, {
116 +                     success: function(test) {
117 +                         // Push was successful
118 +                         console.log("push_successful!");
119 +                     },
120 +                     error: function(error) {
121 +                         // Handle error
122 +                         console.error(error);
123 +                     }
124 +                 });
125 +
126 +                 },
127 +                 error:function (obj, error)
128 +                 {
129 +                     response.error("error!");
130 +                 }
131 +             });
132 +             console.log(obj);
133 +
134 +             // Notify Carers on Alarm
135 +
136 +             }
137 +         },
138 +         error: function(error){
139 +
140 +             }
141 +         });
142 +         //calculate time elapsed since checkin
143 +         }
144 +     },
145 +     error: function(error)
146 +     {
147 +         status.error("ERROR");
148 +     }
149 + });
150 +
151 +
152 +
153 + })
154 +
155 + });
156 +
157 +
158 +
38 159 // Use Parse.Cloud.define to define as many cloud functions as you want.
39 160 // For example:
40 161 Parse.Cloud.define("hello", function(request, response) {
41 162     response.success("Hello world!");
42 163 });
43 164
165 +Parse.Cloud.define("get_user_from_alert", function(request, response){
166 +    //given an alertID, will retrieve user information and stringify
167 +    var alertID = request.params.alertID;
168 +    var alarmquery = new Parse.Query("Alarm");
169 +    var userquery = new Parse.Query("User");
170 +    alarmquery.get(alertID, {
171 +        success: function (alarm_object)      {
172 +            //using this object, we get the username
173 +            var a_username = alarm_object.get("username");
174 +            //console.log(a_username);
175 +            userquery.equalTo("username", a_username);
176 +            userquery.find({
177 +                success: function(user_object) {
178 +                    var json_objects = new Array();
179 +                    json_objects = [{"alarm": alarm_object}, {"user": user_object[0]}];
180 +                    //console.log(json_objects);
181 +                    response.success(JSON.stringify(json_objects));
182 +                },

```

```
183 +         error: function(error)
184 +         {
185 +         }
186 +     }
187 + })
188 +
189 + },
190 + error: function(error)
191 + {
192 + }
193 + }
194 + }
195 + })
196 +});
197 +
44 198 Parse.Cloud.define("get_check_in_list", function(request, response) {
45 199     //THIS FUNCTION gets the table of check ins
46 200     //it then finds when was the last check_in from each user
    ...
93 247     response.success();
94 248 });
95 249
250 +Parse.Cloud.define("dismiss_alarm", function(request, response) {
251 + //given an alarmID, will set its activated state to false.
252 +     var alertID = request.params.alertID;
253 +     var alarmquery = new Parse.Query("Alarm");
254 +     alarmquery.get(alertID, {
255 +         success: function (alarm_object)
256 +         {
257 +             alarm_object.set("Activated", false);
258 +             alarm_object.save(null,
259 +             {
260 +                 success: function()
261 +                 {
262 +                     response.success("yayifications!");
263 +                 },
264 +                 error:function (error)
265 +                 {
266 +                     response.error("error!");
267 +                 }
268 +             });
269 +         },
270 +         error: function (error)
271 +         {
272 +         }
273 +     });
274 + });
275 +
96 276 Parse.Cloud.define("get_active_alarms", function(request, response) {
97 277     //THIS FUNCTION will return an object containing any activated alarms
98 278     //ie. the last message was not a cancel
    ...
112 292     var obj = {}
113 293     var d = new Date();
114 294     var name = results[i].get("Name");
115 295 +     var username = results[i].get("username");
116 296     var inlastalarm = 0;
117 297     d=results[i].updatedAt;
118 298     obj.time = d;
119 299     obj.name = name;
120 300 +     obj.username = username;
121 301     obj.objectid = results[i].id;
122 302     obj.activated = results[i].get("Activated");
123 303
    ...
165 347     },
166 348     error: function(error) {
167 349
168 350 -         reponse.error("something went very terribly wrong");
169 351 +         response.error("something went very terribly wrong");
170 352     }
171 353 })
    ...
```

```
1 +<!DOCTYPE html>
2 +<html>
3 +   <head>
4 +     <% function timeDifference(current, previous) {
5 +
6 +       var msPerMinute = 60 * 1000;
7 +       var msPerHour = msPerMinute * 60;
8 +       var msPerDay = msPerHour * 24;
9 +       var msPerMonth = msPerDay * 30;
10 +      var msPerYear = msPerDay * 365;
11 +
12 +      var elapsed = current - previous;
13 +
14 +      if (elapsed < msPerMinute) {
15 +        return Math.round(elapsed/1000) + ' seconds ago';
16 +      }
17 +
18 +      else if (elapsed < msPerHour) {
19 +        return Math.round(elapsed/msPerMinute) + ' minutes ago';
20 +      }
21 +
22 +      else if (elapsed < msPerDay ) {
23 +        return Math.round(elapsed/msPerHour ) + ' hours ago';
24 +      }
25 +
26 +      else if (elapsed < msPerMonth) {
27 +
28 +        return 'approximately ' + Math.round(elapsed/msPerDay) + ' days ago';
29 +      }
30 +
31 +      else if (elapsed < msPerYear) {
32 +        return 'approximately ' + Math.round(elapsed/msPerMonth) + ' months ago';
33 +      }
34 +
35 +      else {
36 +        return 'approximately ' + Math.round(elapsed/msPerYear ) + ' years ago';
37 +      }
38 +    }
39 +    <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
40 +    <link rel='stylesheet' href='/css/style.css' />
41 +    <title>Active Alarm!</title>
42 +  </head>
43 +  <body>
44 +    <% var date = new Date() %>
45 +    <% date.toISOString() %>
46 +
47 +    <div class="page_title">ALERT INFORMATION</div>
48 +    <p><%= message %></p>
49 +
50 +    <p>
51 +      <% var result = JSON.parse (message) %>
52 +      <%= result[0].alarm.Name %>
53 +      <p><%= url %></p>
54 +
55 +      <table class="list">
56 +        <tr><td class="attr_title">Name</td><td> <%= result[1].user.firstName %> <%= result[1].user.lastName %> </td></tr>
57 +        <tr><td class="attr_title">Email</td><td><%= result[1].user.email %></td></tr>
58 +        <tr><td class="attr_title">Phone Number</td><td><%= result[1].user.phonenumber %></td></tr>
59 +        <tr><td class="attr_title">Alert Status</td><td><%= result[0].alarm.Activated %></td></tr>
60 +
61 +        <tr><td class="attr_title">Alarm triggered</td><td><%= result[0].alarm.createdAt %></td></tr>
62 +      </table>
63 +      <% url += "/dismiss" %>
64 +    </p>
65 +    <a href = <%- url %>>DISMISS ALARM</a>
66 +    <a href = '/css/style.css'>css file</a>
67 +    <a href = "/logout"> Login here</a>
68 +    <a href = "/system"><--- back to active alarms</a>
69 +  </body>
70 +</html>
```

parseCloudCode/cloud/views/hello.ejs

Side-by-side diff

View file

...

```
2 2 <!DOCTYPE html>
3 3 <html>
4 4   <head>
5 5     <title>Sample App</title>
6 6     <title>Group 4 Operator Interface</title>
7 7     <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
```

```
7 + <link rel='stylesheet' href='/css/style.css' />
6 8 </head>
7 9 <body>
8 - <h1>Hello World</h1>
9 10 + <div class="page_title">Welcome to Group 4 - Operator Interface</div>
10 11 <p><%= message %></p>
11 12 <a href ="/login"> Login here</a>
12 13 </body>
```

...

parseCloudCode/cloud/views/login.ejs

Side-by-side diffView file

```
1 1 <html>
2 - <head></head>
2 2 + <head>
3 + <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
4 + <link rel='stylesheet' href='/css/style.css' />
5 + <title>Group 4 Login</title>
6 + </head>
3 7 <body>
4 8
5 9 <form method="post" action="/login">
```

...

parseCloudCode/cloud/views/system.ejs

Side-by-side diffView file

```
3 3 <html>
4 4 <head>
5 5 <script type="text/javascript" src="ejs_production.js"></script>
6 + <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
7 + <link rel='stylesheet' href='/css/style.css' />
6 8 <title>Group4 - Operator Interface</title>
7 9 </head>
8 10 <body>
9 - <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
10 - <link rel='stylesheet' href='/css/style.css' />
11 +
12 <h1>Emergency button Operator System</h1>
13 <p><%= message %></p>
14 <p>
15 <% var result = JSON.parse(message) %>
16
17 <table>
18 + <table class="list">
17 18 <% for(var i=0; i<result.length; i++) {%>
18 19 <tr>
19 20 <td class="messageblock">
20 + <td class="attr_title">
21
22 <%= result[i].name %>
23 </td>
24 <td><%=result[i].time%></td>
25 - <td><% var html = "<a href ='system/alert/'+result[i].objectid+'>"+result[i].objectid+"</a>" %>
26 + <td><% var html = "<a href ='system/alert/'+result[i].objectid+'>"+result[i].objectid+" --> G0</a>" %>
27 <%- html %></td>
```

...

```
29 30 </table>
30 31 </p>
31 32 <a href ='/css/style.css'>css file</a>
32 - <a href ="/login"> Login here</a>
33 + <a href ="/logout"> Logout</a>
34 </body>
35 </html>
```

parseCloudCode/public/css/style.css

Side-by-side diffView file

```
1 1 body {
2 - font-family 'Open Sans', sans-serif; font-weight400;
2 2 + font-family: 'Open Sans', sans-serif;
3 +}
4 +
```

```
5 +.page_title
6 +{
7 +     font-family: 'Open Sans', sans-serif; font-weight:700;
8 +     font-size: 200%
9 +}
10 +
11 +.attr_title
12 +{
13 +     font-family: 'Open Sans', sans-serif; font-weight:700;
14 +     padding:5px;
15 +     text-align:left;
16 +}
17 +
18 +table.list
19 +{
20 +     table-layout: fixed;
21 +     text-align:left;
22 +     background-color:#eeeeee;
23 +     border-radius: 5px;
24 +     border-collapse:collapse;
3 25 }
```

[Blog](#) · [Support](#) · [Plans & pricing](#) · [Documentation](#) · [API](#) · [Server status](#) · [Version info](#) · [Terms of service](#) · [Privacy policy](#)

[JIRA](#) · [Confluence](#) · [Bamboo](#) · [Stash](#) · [SourceTree](#)