```
1   routerApp.controller('listarinvitadosController', function($scope, $http, globalConf) {
2       var newerCursor = "";
3       var stack = [];
4       $scope.guests = [];
5       $scope.search = true;
6
7     $scope.name = "";
8     $scope.lastname = "";
9     $scope.email = "";
10    $scope.company = "";
11
12      $scope.submit = function() {
13        $scope.searching = true;
14        $scope.showTable = false;
15        $scope.search = false;
16        $scope.previousClass="previous disabled";
17        newerCursor="next";
18      $http({
19        method: 'GET',
20        url: globalConf.api_url + "/list?name=" + $scope.name + "&lastname=" + $scope.l
    astname + "&email=" + $scope.email + "&company=" + $scope.company + "&cursor="
21      }).then(function successCallback(response) {
22        $scope.searching = false;
23        $scope.showTable = true;
24        $scope.guests = response.data.guests;
25        newerCursor = response.data.cursor;
26        if($scope.guests.length < globalConf.pageRegs) {
27          $scope.nextClass = "next disabled";
28        } else {
29          $scope.nextClass="next";
30        }
31      }, function errorCallback(response) {
32        $scope.searching = false;
33        $scope.showTable = false;
34        $scope.nextClass="next disabled";
35      });
36    }
37
38      $scope.getNextReport = function() {
39        $scope.searching = true;
40        var data = "/list?name=" + $scope.name + "&lastname=" + $scope.lastname + "&em
    ail=" + $scope.email + "&company=" + $scope.company;
41        if($scope.newerCursor != "") {
42          data += "cursor=" + newerCursor;
43        }
44        $http({
45        method: 'GET',
46        url: globalConf.api_url + data
47      }).then(function successCallback(response) {
48        $scope.searching = false;
49        $scope.showTable = true;
50        $scope.guests = response.data.guests;
51        console.log(response.data.guests);
52        stack.push(newerCursor);
53        newerCursor = response.data.cursor;
54        if($scope.guests.length < globalConf.pageRegs) {
55          $scope.nextClass = "next disabled";
56        }
57        $scope.previousClass="previous";
58      }, function errorCallback(response) {
59        $scope.searching = false;
60        $scope.showTable = false;
61      });
62      };
63
64      $scope.getPreviousReport = function() {
```

```
65        $scope.searching = true;
66        var data = "/list?name=" + $scope.name + "&lastname=" + $scope.lastname + "&em
    ail=" + $scope.email + "&company=" + $scope.company;
67        var olderCursor = stack.pop();
68        if(olderCursor != undefined ∧ stack.length > 0) {
69          data += "cursor=" + olderCursor;
70        } else {
71          $scope.previousClass="previous disabled";
72        }
73        $http({
74        method: 'GET',
75        url: globalConf.api_url + data
76      }).then(function successCallback(response) {
77        $scope.searching = false;
78        $scope.showTable = true;
79        $scope.guests = response.data.guests;
80        newerCursor = response.data.cursor;
81        if($scope.guests.length ≥ globalConf.pageRegs) {
82          $scope.nextClass = "next";
83        }
84      }, function errorCallback(response) {
85        $scope.searching = false;
86        $scope.showTable = false;
87      });
88    }
89  });
```

```
1  routerApp.constant('globalConf' , {
2      'api_url': 'https://app−dot−useful−music−180113.appspot.com',
3      //'api_url': 'http://localhost:8080',
4      'pageRegs': 10
5  });
```

```
1  routerApp.controller('consultarasistenciaController', function($scope, $http, globalConf)
   {
2   $scope.submitDisabled = false;
3   $scope.response = "";
4   $scope.submit = function() {
5     $scope.submitDisabled = true;
6         $http({
7         method: 'GET',
8         url: globalConf.api_url + "/query?id=" + $scope.id,
9         headers: {'Content−Type': 'application/x−www−form−urlencoded'}
10     }).then(function successCallback(response) {
11        $scope.response = "Asistencia confirmada del usuario = Nombre: " + response.data.Name +
      ", Apellido: " + response.data.LastName + ", Email: " + response.data.Email + ", Compania
      : " + response.data.Company;
12     }, function errorCallback(response) {
13        $scope.response = "No existe ninguna confirmacion para el ID: " + $scope.id;
14     });
15     $scope.submitDisabled = false;
16   };
17  });
```

```
1   routerApp.controller('confirmarasistenciaController', function($scope, $http, globalConf)
     {
2     $scope.submitDisabled = false;
3     $scope.response = "";
4     $scope.submit = function() {
5       $scope.submitDisabled = true;
6         var data = {
7               name:       $scope.name,
8               lastname:   $scope.lastname,
9               email:      $scope.email,
10              company:    $scope.company,
11        };
12          $http({
13          method: 'POST',
14          url: globalConf.api_url + "/confirm",
15          params: data,
16          headers: {'Content−Type' : 'application/x−www−form−urlencoded' }
17      }).then(function successCallback(response) {
18        $scope.response = "La confirmacion ha sido exitosa con el ID: " + response.data.id;
19      }, function errorCallback(response) {
20        $scope.response = "Error " + response.status + ":" + response.data;
21      });
22      $scope.submitDisabled = false;
23      };
24  });
```

```
1   // app.js
2   var routerApp = angular.module('routerApp', ['ui.router']);
3
4   routerApp.config(function($stateProvider, $urlRouterProvider) {
5
6       $urlRouterProvider.otherwise('/home');
7
8       $stateProvider
9
10          .state('home', {
11              url: '/home',
12              templateUrl: 'views/home.html'
13          })
14
15          .state('confirmarasistencia', {
16              url: '/confirmarasistencia',
17              templateUrl: 'views/confirmarasistencia.html',
18              controller: 'confirmarasistenciaController'
19          })
20
21          .state('consultarasistencia', {
22              url: '/consultarasistencia',
23              templateUrl: 'views/consultarasistencia.html',
24              controller: 'consultarasistenciaController'
25          })
26
27          .state('listarinvitados', {
28              url: '/listarinvitados',
29              templateUrl: 'views/listarinvitados.html',
30              controller: 'listarinvitadosController'
31          });
32
33  });
```

app.yaml

```
1   runtime: python27
2   api_version: 1
3   threadsafe: true
4
5   handlers:
6   - url: /
7     static_files: www/index.html
8     upload: www/index.html
9
10  - url: /(.*)
11    static_files: www/\1
12    upload: www/(.*)
```

app.yaml

```
1   runtime: go
2   api_version: go1
3   service: app
4
5   handlers:
6   - url: /.*
7     script: _go_app
```

```
1   package app
2
3   import (
4     "fmt"
5     "strconv"
6     "strings"
7     "encoding/json"
8     "net/http"
9       "appengine"
10      "appengine/datastore"
11      "appengine/memcache"
12  )
13
14  // [START guest_struct]
15  type Guest struct {
16      Name    string
17      LastName string
18      Email string
19      Company string
20  }
21  // [END guest_struct]
22
23  // [START list_struct]
24  type List struct {
25      Cursor  string
26      Guests  string
27  }
28  // [END list_struct]
29
30  func init() {
31    http.HandleFunc("/confirm", confirm)
32    http.HandleFunc("/query", query)
33    http.HandleFunc("/list", list)
34  }
35
36  // [START func_confirm]
37  func confirm(w http.ResponseWriter, r *http.Request) {
38      // [START new_context]
39      c := appengine.NewContext(r)
40      // [END new_context]
41      c.Debugf("Entrando a la funcion confirm del servicio default: %v", r.URL)
42      w.Header().Set("Access-Control-Allow-Origin", "*")
43      g := Guest{
44        Name: r.FormValue("name"),
45        LastName: r.FormValue("lastname"),
46        Email: r.FormValue("email"),
47        Company: r.FormValue("company"),
48      }
49
50      //key := datastore.NewIncompleteKey(c, "Guest", datastore.NewKey(c, "Guest",
51      "default_gest", 0, nil))
52      key := datastore.NewIncompleteKey(c, "Guest", nil)
52      id, err := datastore.Put(c, key, &g)
53      if err != nil {
54              http.Error(w, err.Error(), http.StatusInternalServerError)
55              return
56      }
57      stringKey := fmt.Sprintf("%v", id)
58      aStringKey := strings.Split(stringKey, ",")
59    res, _ := json.Marshal(map[string]string{"id": aStringKey[1]})
60      fmt.Fprintf(w, "%s", string(res))
61  }
62  // [END func_confirm]
63
64  // [START func_query]
65  func query(w http.ResponseWriter, r *http.Request) {
```

```
66    var g Guest
67
68      // [START new_context]
69      c := appengine.NewContext(r)
70      w.Header().Set("Access-Control-Allow-Origin", "*")
71      id := r.FormValue("id")
72    n, err := strconv.ParseInt(id, 10, 64)
73    if err != nil {
74        fmt.Fprint(w, "Unable to parse key")
75        return;
76    }
77
78      // Primero busco la entity en la memcache; si no esta,
79      // busco en el datastore
80      _, err1 := memcache.Gob.Get(c, id, &g)
81      c.Debugf("Resultado del cacheo %s", err1)
82      if err1 != nil {
83        c.Debugf("No fue un hit")
84        key := datastore.NewKey(c, "Guest", "", n, nil)
85
86        if err := datastore.Get(c, key, &g); err != nil {
87            http.Error(w, err.Error(), http.StatusInternalServerError)
88            return
89        }
90
91        // Salvo el resultado en la memcache
92      item1 := &memcache.Item{
93        Key:   id,
94        Object: g,
95      }
96      memcache.Gob.Set(c, item1);
97    }
98      // Devuelvo la entity encontrada
99      res, _ := json.Marshal(g)
100     fmt.Fprintf(w, "%s", string(res))
101  }
102  // [END func_query]
103
104  // [START func_list]
105  func list(w http.ResponseWriter, r *http.Request) {
106      // [START new_context]
107      c := appengine.NewContext(r)
108      w.Header().Set("Access-Control-Allow-Origin", "*")
109      // Create a query for all Person entities.
110      const pageSize = 10
111      q := datastore.NewQuery("Guest")
112      if r.FormValue("name") != "" {
113        q = q.Filter("Name =", r.FormValue("name"))
114      }
115      if r.FormValue("lastname") != "" {
116        q = q.Filter("LastName =", r.FormValue("lastname"))
117      }
118      if r.FormValue("email") != "" {
119        q = q.Filter("Email =", r.FormValue("email"))
120      }
121      if r.FormValue("company") != "" {
122        q = q.Filter("Company =", r.FormValue("company"))
123      }
124
125      q = q.Limit(pageSize)
126
127      // If the application stored a cursor during a previous request, use it.
128      if r.FormValue("cursor") != "" {
129          cursor, err := datastore.DecodeCursor(r.FormValue("cursor"))
130          if err == nil {
131              q = q.Start(cursor)
```

app.go

```
132              }
133      }
134      // Iterate over the results.
135      var guests []string
136
137      t := q.Run(c)
138      for {
139              var g Guest
140              id, err := t.Next(&g)
141              if err == datastore.Done {
142                      break
143              }
144              if err != nil {
145                      c.Errorf("fetching next Guest: %v", err)
146                      break
147              }
148
149              stringKey := fmt.Sprintf("%v", id)
150              aStringKey := strings.Split(stringKey, ",")
151
152              res, _ := json.Marshal(g)
153              guests = append(guests, "{\"id\":\"" + aStringKey[1] + "\", \"guest\":
154      " + string(res) + "}")
155      }
156      // Get updated cursor and store it for next time.
157      strGuests := strings.Join(guests, ",")
158      c.Debugf("fetching next Guest: %s", strGuests)
159      if cursor, err := t.Cursor(); err == nil {
160        fmt.Fprintf(w, "%s", "{\"cursor\": \"" + cursor.String() +"\", \"guests\":["
         + strGuests +"]}")
161      }
162 }
163 // [END func_list]
```

Table of Content

Table of Contents