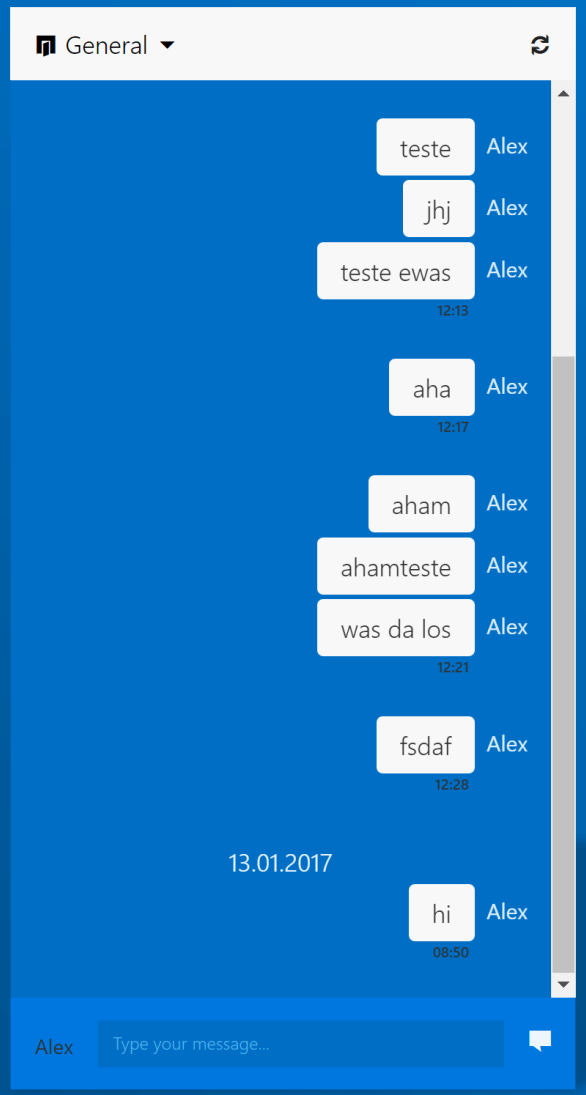
Alexandru Gogan



Chatin.

A simple single page chat using

ASP.NET 4.5 Web API & AngularJS 1.5

https://github.com/alx-andru/Chatin

Chatin.

Alexandru Gogan

Features

- Basic messaging
- Switch between chatrooms
- Refresh list of messages
- Show text, timestamp and username
- Group messages within a 3min time period

Technical Requirements

- ASP.NET 4.5 or 4.6
- AngularJS 1.5

Alexandru Gogan

Framework Overview

.NET Web API

- CORS
- EntityFramework
- SQLite

Front-End

- angular packages (core,animate,resource,sanitize)
- momentJS (+ angular-moment)
- Underscore
- UI Office Fabric

Alexandru Gogan

AngularJS

Chatin.

ng-app="ChatinApp"

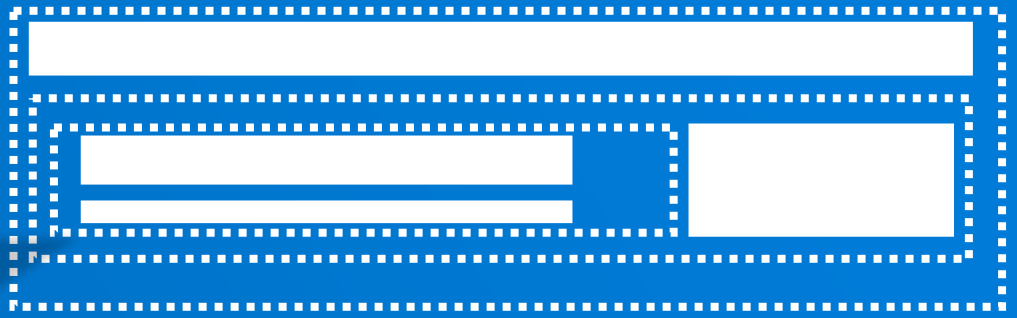
ng-controller="ChatinController"

<div class="chatbox__header”

<messages-box ...>

<div class="chatbox__actions”

Chatin.



```
3   <div class="chatbox-message" ng-repeat="message in messages track by $index">
4       {{previous = ($index-1)>=0?$index-1:0; ""}}
5       <div class="chatbox-message__date"
6            ng-if="(message.timestamp |  amDateFormat:'DD.MM.YYYY') !== (messages[previous].timestamp | amDateFormat: 'DD.MM.YYYY')
7            || ($index-1)==-1">
8           {{message.timestamp | amDateFormat:'DD.MM.YYYY'}}
9       </div>
10
11      <div class="chatbox-message-item" ng-class="{'chatbox-message-item--right':isUser($index)}">
12          <div>
13              <span class="chatbox-message-item__text"
14                    ng-class="{'chatbox-message-item__text--right':isUser($index)}">
15                  {{message.text}}
16              </span>
17              <span class="chatbox-message-item__time ms-font-mi"
18                    ng-class="{'chatbox-message-item__time--right':isUser($index)}"
19                    ng-if="displayTime($index)">
20                  {{message.timestamp | amDateFormat:'HH:mm'}}
21              </span>
22          </div>
23          <span class="chatbox-message-item__author ms-font-m-plus"
24                ng-class="{'chatbox-message-item__author--right':isUser($index)}">{{message.user.name}}
25          </span>
26      </div>
27  </div>
```

Alexandru Gogan

```
10  // Base URL for REST-API
11  var url = 'http://localhost:51852';
12  chatin.factory('Messages', function ($resource, $window) {
13      return $resource(url + '/api/room/:room/messages', {
14          room: '@roomid',
15      }, {
16          inRoom: {
17              method: 'GET',
18              isArray: true,
19              params: {
20                  roomid: 'string'
21              }
22          }
23      }
24      );
25
26  });
```

Chatin.

Alexandru Gogan

```
28 chatin.factory('Rooms', function ($resource, $window) {
29     return $resource(url + '/api/room');
30 });
31
32 chatin.factory('Users', function ($resource, $window) {
33     return $resource(url + '/api/user');
34 });
35
36 chatin.factory('_', function ($window) {
37     // Helper mixin to sort by keys and not just attributes
38     $window._.mixin({
39         'sortKeysBy': function (obj, comparator) {
40             var keys = _.sortBy(_.keys(obj), function (key) {
41                 return comparator ? comparator(obj[key], key) : key;
42             });
43
44             return _.object(keys, _.map(keys, function (key) {
45                 return obj[key];
46             }));
47         }
48     });
49     return $window._;
50 });
```

Alexandru Gogan

```
52 chatin.directive('messagesBox', function ($timeout) {
53     return {
54         scope: {
55             messages: '=messages',
56             user: '=user',
57             timestamps: '=timestamps'
58         },
59         templateUrl: '../Template/Message',
60         link: function (scope, element) {
61             scope.isUser = function(idx){
62                 console.log('user');
63                 return scope.messages[idx].user.name === scope.user.name;
64             };
65 
66             // determine if messages can be grouped together
67             // show only one timestamp for all the messages within 3min
68             scope.displayTime = function (idx) {
69                 return _.find(scope.timestamps, function (item) {
70                     return item == idx;
71                 });
72 
73             };
74 
75             scope.$watchCollection(scope.messages, function (newVal) {
76                 // delay scroll to ensure view is rendered
77                 $timeout(function () {
78                     element[0].scrollTop = element[0].scrollHeight;
79                 },100);
80             });
81 
82         }
83     }
84 });
```

Alexandru Gogan

```
105     function determineActiveTimestamps(messages) {
106         // reset timestamps
107         $scope.displayTimestamps = [];
108         var currentGroup;
109 
110         for (var msgIdx = messages.length - 1; msgIdx >= 0; msgIdx--) {
111             var message = messages[msgIdx];
112             var timestamp = moment(message.timestamp);
113 
114             // init for first iteration
115             if (currentGroup == undefined) {
116                 currentGroup = timestamp;
117                 $scope.displayTimestamps.push(msgIdx);
118             }
119 
120             // either fits in the group
121             if (currentGroup.diff(timestamp, 'minutes') <= 3) {
122 
123             } else {
124                 // or create a new one
125                 currentGroup = timestamp;
126                 $scope.displayTimestamps.push(msgIdx);
127             }
128         }
129     }
```

Alexandru Gogan

```
131     // Toggle Rooms
132     $scope.isOpen = false;
133
134     $scope.joinRoom = function (idx) {
135         $scope.activeRoom = $scope.rooms[idx];
136         $scope.refresh();
137     };
138
139     $scope.toggleMenu = function () {
140         $scope.isOpen = !$scope.isOpen;
141     }
142
143     // Refresh message list
144     $scope.refresh = function () {
145         // Messages
146         $scope.messages = Messages.inRoom({
147             room: $scope.activeRoom.id,
148         }, function (messages) {
149             determineActiveTimestamps(messages);
150         });
151     };
152
153     // find current active user in list and return next.
154     $scope.toggleUser = function () {
155         var position = 0;
156         for (var user in $scope.users) {
157             if ($scope.users[user] === $scope.user) {
158                 position = parseInt(user) + 1;
159             }
160         };
161         // user was last in list, switch to first user
162         var nextUser = position == $scope.users.length ? 0 : position;
163         $scope.user = $scope.users[nextUser];
164     }
```

Alexandru Gogan

```
169     // Textinput of new message
170     $scope.msg = '';
171
172     // Send the message
173     $scope.send = function () {
174         var text = $scope.msg;
175         // if empty, ignore it
176         if (text.length == 0) {
177             return;
178         }
179
180         var message = new Messages();
181         message.text = text;
182         message.userid = $scope.user.id;
183         message.roomid = $scope.activeRoom.id;
184
185         message.$save().then(function () {
186             // reload list to display new message
187             $scope.refresh();
188         });
189     }
```

Alexandru Gogan

.NET

Chatin.

Alexandru Gogan

```
6     public class BundleConfig
7     {
8         public static void RegisterBundles(BundleCollection bundles)
9         {
10            bundles.Add(new ScriptBundle("~/Scripts/chatin").Include(
11                "~/Scripts/app/ChatinController.js"
12            ));
13
14            bundles.Add(new ScriptBundle("~/Scripts/angular").Include(
15                "~/Scripts/angular.min.js",
16                "~/Scripts/angular-sanitize.min.js",
17                "~/Scripts/angular-resource.min.js",
18                "~/Scripts/ngOfficeUiFabric.min.js",
19                "~/Scripts/moment.min.js",
20                "~/Scripts/angular-moment.min.js",
21                "~/Scripts/underscore.min.js"
22            ));
23
24            bundles.Add(new StyleBundle("~/Content/css").Include(
25                "~/Content/app/chatin.css"
26            ));
27
28            bundles.Add(new StyleBundle("~/Content/fabric").Include(
29                "~/Content/fabric.min.css",
30                "~/Content/fabric.components.min.css"
31            ));
32        }
33    }
34 }
```

Alexandru Gogan

```
8   namespace Chatin
9   {
10      public class RouteConfig
11      {
12          public static void RegisterRoutes(RouteCollection routes)
13          {
14              routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
15
16              routes.MapRoute(
17                  name: "Default",
18                  url: "{controller}/{action}/{id}",
19                  defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
20              );
21          }
22
23      }
24  }
```

Alexandru Gogan

```
5   namespace Chatin.Models
6   {
7       public class Message
8       {
9           [Key]
10          public int Id { get; set; }
11
12          public String Text { get; set; }
13          public DateTime Timestamp { get; set; }
14
15          //[ForeignKey("User")]
16          public int? UserID { get; set; }
17          public virtual User User { get; set; }
18
19          //[ForeignKey("Room")]
20          public int? RoomID { get; set; }
21          public virtual Room Room { get; set; }
22      }
23  }
```

Alexandru Gogan

```
4   namespace Chatin.Models
5   {
6       public class Room
7       {
8
9           [Key]
10          public int Id { get; set; }
11          public string Name { get; set; }
12          public string Icon { get; set; }
13
14          public virtual ICollection<User> Users { get; set; }
15          public virtual ICollection<Message> Messages { get; set; }
16
17      }
18  }
```

Alexandru Gogan

```
4   namespace Chatin.Models
5   {
6       public class User
7       {
8           [Key]
9           public int Id { get; set; }
10          public string Name { get; set; }
11      }
12  }
```

Alexandru Gogan

```
5   namespace Chatin.DAL
6   {
7       public class ChatinContext : DbContext
8       {
9           public ChatinContext() : base("ChatinContext")
10          {
11
12          }
13
14          public DbSet<Message> Messages { get; set; }
15          public DbSet<User> Users { get; set; }
16          public DbSet<Room> Rooms { get; set; }
17      }
18
19
20  }
```

Alexandru Gogan

```
10      public class ChatinInitializer : System.Data.Entity.DropCreateDatabaseIfModelChanges<ChatinContext>
11      {
12          protected override void Seed(ChatinContext context)
13          {
14              var users = new List<User>
15              {
16                  new User {Name="Addo" },
17                  new User {Name="Alex" },
18                  new User {Name="Miro" }
19              };
20              users.ForEach(u => context.Users.Add(u));
21              context.SaveChanges();
22
23              var rooms = new List<Room>
24              {
25                  new Room {Name="General", Icon="room" },
26                  new Room {Name="Private", Icon="lock"}
27
28              };
29              rooms.ForEach(r => context.Rooms.Add(r));
30              context.SaveChanges();
```

Chatin.

Alexandru Gogan

```
9   namespace Chatin.Controllers
10  {
11      [RoutePrefix("api/message")]
12      public class MessageController : ApiController
13      {
14          public ChatinContext db = new ChatinContext();
15
16
17          public MessageController()
18          {
19
20          }
21          [Route("")]
22          public IEnumerable<Message> GetAllMessages()
23          {
24              return db.Messages.ToList();
25          }
26          [HttpPost]
27          [Route("")]
28          public IHttpActionResult Add([FromBody] Message message)
29          {
30              message.Timestamp = DateTime.Now;
31              db.Messages.Add(message);
32              db.SaveChanges();
33              return Ok(db.Messages.ToList());
34          }
35
36          [Route("{id:int}")]
37          public IHttpActionResult GetMessage(int id)
38          {
39              var message = db.Messages.FirstOrDefault((msg) => msg.Id == id);
40              if (message == null)
41              {
42                  return NotFound();
43              }
44              return Ok(message);
45          }
46
```

```
47          protected override void Dispose(bool disposing)
48          {
49              db.Dispose();
50              base.Dispose(disposing);
51          }
52
```

Alexandru Gogan

Chatin.

https://github.com/alx-andru/Chatin