

LAMPIRAN

Script

CSS

pageslider.css

```
#container {
    position: absolute;
    width: 100%;
    height: 100%;
    overflow: hidden;
}

.page {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    -webkit-transform: translate3d(0, 0, 0);
    transform: translate3d(0, 0, 0);
}

.page.page-left {
    -webkit-transform: translate3d(-100%, 0, 0);
    transform: translate3d(-100%, 0, 0);
}

.page.page-center {
```

```
-webkit-transform: translate3d(0, 0, 0);  
transform: translate3d(0, 0, 0);  
}
```

```
.page.page-right {  
  -webkit-transform: translate3d(100%, 0, 0);  
  transform: translate3d(100%, 0, 0);  
}
```

```
.page.transition {  
  -webkit-transition-duration: .25s;  
  transition-duration: .25s;  
}
```

styles.css

```
.topcoat-navigation-bar {  
  z-index: 10;  
}
```

```
.scroller {  
  overflow: auto;  
  -webkit-overflow-scrolling: touch;  
  position: absolute;  
  top: 168px;  
  bottom: 0px;  
  left: 0px;  
  right: 0px;  
}
```

```
.topcoat-icon--back {  
  background: url("images/back_light.svg") no-repeat;  
  -webkit-background-size: cover;  
  -moz-background-size: cover;  
  background-size: cover;  
}
```

```
.topcoat-list__container {  
  border-bottom: none;  
  clear: both;  
}
```

```
.count {  
  color: #58C0FF;  
  position: absolute;  
  right: 50px;  
  top: 26px;  
  font-weight: bold;  
}
```

```
.search-bar {  
  padding: 12px 10px 12px 8px;  
}
```

```
a {  
  text-decoration: none;  
  color: inherit;  
  -webkit-touch-callout: none;  
  -webkit-tap-highlight-color: rgb(0, 0, 0);  
}
```

```
}
```

```
.details {  
  margin: auto;  
}
```

```
.details>img {  
  float:left;  
  margin:10px;  
  width: 114px;  
  height: 80px;  
}
```

```
.details h1 {  
  padding: 12px 0px 4px 0px;  
  margin: 0px 0px 0px 0px;  
  font-size: 1.2rem;  
  color: #C6C8C8;  
}
```

```
.details h2 {  
  padding: 0px 0px 0px 0px;  
  margin: 0px 0px 0px 0px;  
  font-size: 1.1rem;  
  font-weight: normal;  
  color: #888;  
}
```

```
.list li {
```

```
position: relative;
text-align: left;
clear: both;
padding: 12px 60px 16px 8px !important;
}
```

```
.list > li p:nth-of-type(1) {
margin: 0px;
padding-top: 4px;
font-weight: bold;
}
```

```
.list > li p:nth-of-type(2) {
margin: 0px;
color: #888;
}
```

```
.list > li img {
width: 69px;
height: 48px;
float: left;
margin-top: 0px;
margin-right: 8px;
}
```

```
.list li:active {
background-color: #404141;
-webkit-box-shadow: inset 0 1px rgba(0,0,0,0.18);
box-shadow: inset 0 1px rgba(0,0,0,0.18);
}
```

```
}
```

```
.action-list > li {  
  position: relative;  
}
```

```
.action-icon {  
  position: absolute !important;  
  top: 22px;  
  right: 20px !important;  
  width: 28px !important;  
  height: 28px;  
}
```

```
.actions li p:nth-of-type(2) {  
  color: #58C0FF ;  
}
```

```
ul {  
  clear: both;  
  border-top: none !important;  
}
```

```
.icon-call {  
  background: transparent url(images/call.svg);  
  background-repeat: no-repeat;  
  -webkit-background-size: cover;  
  -moz-background-size: cover;  
  background-size: cover;
```

```
}
```

```
.icon-sms {  
  background: transparent url(images/chat.svg);  
  background-repeat: no-repeat;  
  -webkit-background-size: cover;  
  -moz-background-size: cover;  
  background-size: cover;  
}
```

```
.icon-mail {  
  background: transparent url(images/email.svg);  
  background-repeat: no-repeat;  
  -webkit-background-size: cover;  
  -moz-background-size: cover;  
  background-size: cover;  
}
```

```
.icon-manager {  
  background: transparent url(images/next.svg);  
  background-repeat: no-repeat;  
  -webkit-background-size: cover;  
  -moz-background-size: cover;  
  background-size: cover;  
}
```

```
.icon-reports {  
  background: transparent url(images/next.svg);  
  background-repeat: no-repeat;
```

```
-webkit-background-size: cover;
-moz-background-size: cover;
background-size: cover;
}

.icon-location {
  background: transparent url(images/location.svg);
  background-repeat: no-repeat;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  background-size: cover;
}

.chevron {
  background: transparent url(images/next_blue.svg);
  background-repeat: no-repeat;
  background-size: contain;
  width: 20px;
  height: 20px;
  position: absolute;
  right: 12px;
  top: 22px;
  height: 50px;
  width: 28px;
}

#map {
  position: absolute;
  top: 64px;
```



```
left: 0;
right: 0;
bottom: 0;
overflow: hidden;
}
```

```
app.js
var app = {
  views: {},
  models: {},
  routers: {},
  utils: {},
  adapters: {}
};
```

```
$(document).on("ready", function () {
  app.router = new app.routers.AppRouter();
  app.utils.templates.load(["HomeView", "EmployeeView",
"EmployeeListItemView", "ReportsView", "MapView"],
  function () {
    app.router = new app.routers.AppRouter();
    Backbone.history.start();
  });
});
```

Java Script

memory-adapter.js

```
app.adapters.employee = (function () {

    var initialize = function() {
        // No Initialization required
        var deferred = $.Deferred();
        deferred.resolve();
        return deferred.promise();
    },

    findById = function (id) {
        var deferred = $.Deferred();
        var employee = null;
        var l = employees.length;
        for (var i=0; i < l; i++) {
            if (employees[i].id === id) {
                employee = employees[i];
                break;
            }
        }
        deferred.resolve(employee);
        return deferred.promise();
    },

    findByName = function(searchKey) {
        var deferred = $.Deferred();
        var results = employees.filter(function(element) {
            var fullName = element.namaTempat + " " + element.namaPic;
```

```

    return fullName.toLowerCase().indexOf(searchKey.toLowerCase()) > -
1;
});
deferred.resolve(results);
return deferred.promise();
},

```

```

    findByManager = function (linkLokasi) {
    var deferred = $.Deferred();
    var results = employees.filter(function (element) {
        return linkLokasi === element.linkLokasi;
    });
    deferred.resolve(results);
    return deferred.promise();
},

```

```

    employees = [
{"id": 1, "namaTempat": "Hotel", "namaPic": "hotel", "linkLokasi": 0,
"namaLokasi": "", "reports": 2, "title": "Daftar Hotel di DKI Jakarta",
"officePhone": "-", "alamat": "-", "city": "DKI Jakarta", "website": "-"},

{"id": 2, "namaTempat": "Rumah Sakit", "namaPic": "hospital", "linkLokasi":
0, "namaLokasi": "", "reports": 2, "title": "Daftar Rumah Sakit di DKI Jakarta",
"officePhone": "-", "alamat": "-", "city": "DKI Jakarta", "website": "-"},

{"id": 3, "namaTempat": "Tempat Wisata", "namaPic": "wisata", "linkLokasi":
0, "namaLokasi": "", "reports": 2, "title": "Daftar Tempat Wisata di DKI
Jakarta",

```

"officePhone": "-", "alamat": "-", "city": "DKI Jakarta", "website": "-"},

{"id": 4, "namaTempat": "Aryaduta Semanggi", "namaPic": "aryaduta",
"linkLokasi": 1, "namaLokasi": "Hotel", "reports": "details", "title": "Jakarta
Pusat",

"officePhone": "02123521234", "alamat": "Jalan Garnisun Dalam No. 8, 12930",
"city": "Jakarta Pusat", "website": "www.aryaduta.com", "latlong": "-6.21999,
106.81574"},

{"id": 5, "namaTempat": "Ciputra Hotel", "namaPic": "ciputra", "linkLokasi": 1,
"namaLokasi": "Hotel", "reports": "details", "title": "Jakarta Barat",

"officePhone": "0215660640", "alamat": "Jalan Letnan Jenderal S Parman,
11470", "city": "Jakarta Barat", "website": "www.ciputra.com", "latlong": "-
6.20876, 106.84560"},

{"id": 6, "namaTempat": "RS Pondok Indah", "namaPic": "rspi", "linkLokasi":
2, "namaLokasi": "Rumah Sakit", "reports": "details", "title": "Jakarta Selatan",

"officePhone": "0217805296", "alamat": "Jalan Metro Pondok Indah, 12310",
"city": "Jakarta Selatan", "website": "www.rspondokindah.co.id", "latlong": "-
6.18673, 106.74862"},

{"id": 7, "namaTempat": "RS Harapan Kita", "namaPic": "harapan",
"linkLokasi": 2, "namaLokasi": "Rumah Sakit", "reports": "details", "title":
"Jakarta Pusat",

"officePhone": "0215668284", "alamat": "Jalan Letjen. S. Parman No.Kav 71,
11420", "city": "Jakarta Pusat", "website": "www.rsabhk.co.id", "latlong": "-
6.18470, 106.79873"},

```
{ "id": 8, "namaTempat": "Taman Impian Jaya Ancol", "namaPic": "ancol",  
"linkLokasi": 3, "namaLokasi": "Tempat Wisata", "reports": "details", "title":  
"Jakarta Utara",  
"officePhone": "0216453456", "alamat": "Jalan Pantai Indah, 14430", "city":  
"Jakarta Utara", "website": "www.ancol.com", "latlong": "-6.12353,  
106.83187"},
```

```
{ "id": 9,
```

```
"namaTempat": "Taman Mini Indonesia Indah",  
"namaPic": "tmii",  
"linkLokasi": 3,  
"namaLokasi": "Tempat Wisata",  
"reports": "details",  
"title": "Jakarta Timur",  
"officePhone": "0218401687",  
"alamat": "Jalan TMII Cipayung, 13560A",  
"city": "Jakarta Timur",  
"website": "www.tamanmini.com",  
"latlong": "-6.30245, 106.89516"},
```

```
{ "id": 10,
```

```
"namaTempat": "Pullman Central Park",  
"namaPic": "pullman",  
  
"linkLokasi": 1,  
"namaLokasi": "Hotel",  
"reports": "details",
```

"title": "Jakarta Barat",
"officePhone": "02129200088",
"alamat": "Jalan Letnan Jendral, S. Parman, 11470",
"city": "Jakarta Barat",
"website": "www.pullmanjakartacentralpark.com",
"latlong": "-6.17513, 106.78539"},

{"id": 11,

"namaTempat": "RS Mitra Keluarga Kelapa Gading",
"namaPic": "mitrakeluarga",

"linkLokasi": 2,
"namaLokasi": "Rumah Sakit",
"reports": "details",
"title": "Jakarta Utara",
"officePhone": "02145852700",
"alamat": "Jalan Bukit Gading Raya Kav. 2 Kelapa Gading, 14240",
"city": "Jakarta Utara",
"website": "www.mitrakeluarga.com/gading/",
"latlong": "-6.15343, 106.89729"},

{"id": 12,

"namaTempat": "RS Siloam Hospital",
"namaPic": "siloam",

"linkLokasi": 2,
"namaLokasi": "Rumah Sakit",

"reports": "details",
"title": "Jakarta Barat",
"officePhone": "0215300888",
"alamat": "Jalan Raya Perjuangan Kav 8 Kebon Jeruk, 11530",
"city": "Jakarta Barat",
"website": "www.siloamhospitals.com",
"latlong": "-6.23463, 106.75295"},

{"id": 13,

"namaTempat": "Novotel Mangga Dua",
"namaPic": "novotel",

"linkLokasi": 1,
"namaLokasi": "Hotel",
"reports": "details",
"title": "Jakarta Utara",
"officePhone": "02162312800",
"alamat": "Jalan Gunung Sahari Raya, 14420",
"city": "Jakarta Utara",
"website": "www.novotel.com",
"latlong": "-6.16174, 106.83868"},

{"id": 14,

"namaTempat": "Merlynn Park",
"namaPic": "merlynn",

"linkLokasi": 1,

```
"namaLokasi": "Hotel",  
"reports": "details",  
"title": "Jakarta Pusat",  
"officePhone": "02130026888",  
"alamat": "Jalan K.H. Hasyim Ashari 29-31, Gajah Mada",  
"city": "Jakarta Pusat",  
"website": "www.merlynnparkhotel-jakarta.com/",  
"latlong": "-6.20876 106.84560"},
```

```
{"id": 15,
```

```
"namaTempat": "RS Harapan Bunda",  
"namaPic": "harapanbunda",  
  
"linkLokasi": 2,  
"namaLokasi": "Rumah Sakit",  
"reports": "details",  
"title": "Jakarta Timur",  
"officePhone": "0218400257",  
"alamat": "Jalan Raya Bogor Km 22 No. 24 Ciracas, 13750",  
"city": "Jakarta Timur",  
"website": "www.rsharapanbunda.com/",  
"latlong": "-6.31554, 106.86799"},
```

```
{"id": 16,
```

```
"namaTempat": "Taman Margasatwa Ragunan",  
"namaPic": "ragunan",
```


"linkLokasi": 3,
"namaLokasi": "Tempat Wisata",
"reports": "details",
"title": "Jakarta Selatan",
"officePhone": "0217805280",
"alamat": "Pasar Minggu, Kota Jakarta Selatan, DKI Jakarta",
"city": "Jakarta Selatan",
"website": "www.ragunanzoo.jakarta.go.id/",
"latlong": "-6.29398, 106.82374"},

{"id": 17,

"namaTempat": "Museum Kota Tua",
"namaPic": "kotatua",

"linkLokasi": 3,
"namaLokasi": "Tempat Wisata",
"reports": "details",
"title": "Jakarta Barat",
"officePhone": "0216916275",
"alamat": "Jalan Taman Fatahillah No 2, Jakarta Barat",
"city": "Jakarta Barat",
"website": "www.kotatuajakarta.org",
"latlong": "-6.13520, 106.81330"},

{"id": 18,

"namaTempat": "Monumen Nasional",

"namaPic": "monas",

"linkLokasi": 3,

"namaLokasi": "Tempat Wisata",

"reports": "details",

"title": "Jakarta Pusat",

"officePhone": "0213822255",

"alamat": "Medan Merdeka, Jakarta Pusat, Indonesia",

"city": "Jakarta Pusat",

"website": "www.jakarta.go.id/web/encyclopedia/detail/1989/Nasional-Monumen",

"latlong": "-6.17621, 106.82304"},

{ "id": 19,

"namaTempat": "TamaN Anggrek",

"namaPic": "monas",

"linkLokasi": 3,

"namaLokasi": "Tempat Wisata",

"reports": "details",

"title": "Jakarta Pusat",

"officePhone": "0213822255",

"alamat": "Medan Merdeka, Jakarta Pusat, Indonesia",

"city": "Jakarta Pusat",

"website": "www.jakarta.go.id/web/encyclopedia/detail/1989/Nasional-Monumen",

"latlong": "-6.17621, 106.82304"},

```
];

// The public API
return {
  findById: findById,
  findByName: findByName,
  findByManager: findByManager
};

}());
```

EmployeeModel.js

```
app.models.Employee = Backbone.Model.extend({

  initialize: function () {
    this.reports = new app.models.ReportsCollection();
    this.reports.parent = this;
  },

  sync: function(method, model, options) {
    if (method === "read") {
      app.adapters.employee.findById(parseInt(this.id)).done(function (data) {
        options.success(data);
      });
    }
  }

});
```

```
app.models.EmployeeCollection = Backbone.Collection.extend({

  model: app.models.Employee,

  sync: function(method, model, options) {
    if (method === "read") {
      app.adapters.employee.findByName(options.data.name).done(function
(data) {
        options.success(data);
      });
    }
  }
});
```

```
app.models.ReportsCollection = Backbone.Collection.extend({

  model: app.models.Employee,

  sync: function(method, model, options) {
    if (method === "read") {
      console.log("find by manager");
      app.adapters.employee.findByManager(this.parent.id).done(function
(data) {
        options.success(data);
      });
    }
  }
});
```

```
});
```

AppRouter.js

```
app.routers.AppRouter = Backbone.Router.extend({

  routes: {
    "": "home",
    "employees/:id": "employeeDetails",
    "employees/:id/reports": "reports",
    "employees/:id/map": "map"
  },

  initialize: function () {
    app.slider = new PageSlider($('body'));
  },

  home: function () {
    // Since the home view never changes, we instantiate it and render it only
once
    if (!app.homeView) {
      app.homeView = new app.views.HomeView();
      app.homeView.render();
    } else {
      console.log('reusing home view');
      app.homeView.delegateEvents(); // delegate events when the view is
recycled
    }
    app.slider.slidePage(app.homeView.$el);
  }
});
```

```
},
```

```
employeeDetails: function (id) {
  var employee = new app.models.Employee({id: id});
  employee.fetch({
    success: function (data) {
      // Note that we could also 'recycle' the same instance of
      EmployeeFullView
      // instead of creating new instances
      app.slider.slidePage(new app.views.EmployeeView({ model:
data}).render().$el);
    }
  });
},
```

```
reports: function (id) {
  var employee = new app.models.Employee({id: id});
  employee.fetch({
    success: function (data) {
      // Note that we could also 'recycle' the same instance of
      EmployeeFullView
      // instead of creating new instances
      app.slider.slidePage(new app.views.ReportsView({ model:
data}).render().$el);
    }
  });
},
```

```
map: function (id) {
```

```

    app.slider.slidePage(new app.views.MapView().render().$el);
  }

});

```

Templates.js

```

app.utils.templates = (function() {

  var load = function(views, callback) {

    var deferreds = [];

    $.each(views, function(index, view) {
      if (app.views[view]) {
        deferreds.push($.get('tpl/' + view + '.html', function(data) {
          app.views[view].prototype.template = _template(data);
        }, 'html'));
      } else {
        alert(view + " not found");
      }
    });

    $.when.apply(null, deferreds).done(callback);
  }

  // The public API
  return {
    load: load
  }

```

```
};
```

```
})();
```

EmployeeListView.js

```
app.views.EmployeeListView = Backbone.View.extend({

  tagName:'ul',

  attributes: {class: 'topcoat-list list'},

  initialize:function () {
    var self = this;
    this.model.on("reset", this.render, this);
    this.model.on("add", function (employee) {
      self.$el.append(new
app.views.EmployeeListItemView({model:employee}).render().el);
    });
  },

  render:function () {
    this.$el.empty();
    _.each(this.model.models, function (employee) {
      this.$el.append(new
app.views.EmployeeListItemView({model:employee}).render().el);
    }, this);
    return this;
  }
});
```



```
app.views.EmployeeListItemView = Backbone.View.extend({

  tagName:"li",

  className:"topcoat-list__item",

  initialize:function () {
    this.model.on("change", this.render, this);
    this.model.on("destroy", this.close, this);
  },

  render:function () {
    this.$el.html(this.template(this.model.attributes));
    return this;
  }

});
```

EmployeeView.js

```
app.views.EmployeeView = Backbone.View.extend({

  render: function () {
    this.$el.html(this.template(this.model.attributes));
    return this;
  },

  events: {
    "click .back-button": "back",
```

```

        "click .add-location-btn": "addLocation"
    },

    addLocation: function(event) {
    event.preventDefault();
    console.log('addLocation');
    navigator.geolocation.getCurrentPosition(
        function(position) {
            $('.location', this.el).html(position.coords.latitude + '
+position.coords.longitude);
        },
        function() {
            alert('Error getting location');
        });
    return false;
    },

    back: function(event) {
        window.history.back();
        return false;
    }

});

```

HomeView.js

```

app.views.HomeView = Backbone.View.extend({

    initialize: function () {
        this.searchResults = new app.models.EmployeeCollection();
    }
});

```

```
    this.searchresultsView = new app.views.EmployeeListView({ model:
this.searchResults });
  },

  render: function () {
    this.$el.html(this.template());
    $('.scroller', this.el).append(this.searchresultsView.render().el);
    return this;
  },

  events: {
    "keyup .search-key": "search",
    "keypress .search-key": "onkeypress"
  },

  search: function (event) {
    var key = $('.search-key').val();
    this.searchResults.fetch({reset: true, data: {name: key}});
  },

  onkeypress: function (event) {
    if (event.keyCode === 13) { // enter key pressed
      event.preventDefault();
    }
  }
});
```

MapView.js

```

app.views.MapView = Backbone.View.extend({

  render: function () {
    this.$el.html(this.template());
    setTimeout(function() {
      // create a map in the "map" div, set the view to a given place and zoom
      var map = L.map('map', {zoomControl:false}).setView([-6.21999,
106.81574], 16);
      // add an OpenStreetMap tile layer
      L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
        attribution:          '&copy;          <a
href="http://osm.org/copyright">OpenStreetMap</a> contributors'
      }).addTo(map);

      // add a marker in the given location, attach some popup content to it and
open the popup
      L.marker([-6.21999, 106.81574]).addTo(map);
    });

    return this;
  },

  events: {
    "click .back-button": "back"
  },

  back: function() {
    window.history.back();
    return false;
  }
});

```

```

    }

```

```

});

```

ReportsView.js

```

app.views.ReportsView = Backbone.View.extend({

```

```

  initialize: function () {

```

```

    this.store = this.options.store;

```

```

  },

```

```

  render: function () {

```

```

    this.$el.html(this.template(this.model.attributes));

```

```

    this.model.reports.fetch();

```

```

    $(".scroller", this.el).html(new app.views.EmployeeListView({ model:
this.model.reports }).render().el);

```

```

    return this;

```

```

  },

```

```

  events: {

```

```

    "click .back-button": "back"

```

```

  },

```

```

  back: function() {

```

```

    window.history.back();

```

```

    return false;

```

```

  }

```

```

});

```