

## In site.js

// LOAD AVAILABLE COUNTRIES WITH acled\_sample.api

1. Make an AJAX query to get the list of available countries

```
$('#country-selector').addClass('loading');  
var countries_ajax = acled_sample.api.getCountries();
```

2. Wait for AJAX query and populate the dropdown menu

```
$.when(countries_ajax).then(function(countries) {  
    $('#country-selector').removeClass('loading');  
    countries.sort();  
    countries.forEach(function(c) {  
        $('#country-selector').append("<option value=\"" + c + "\">" + c + "</option>");  
    });  
});
```

// ADD ON CHANGE EVENT TO COUNTRY SELECTOR --> initDashboard

1. Get selected country and make & wait an AJAX query to get the events

```
$('#loader-container').addClass('loading').show();  
  
var country = $('#country-selector').val();  
var events_ajax = acled_sample.api.getEventsInCountry(country);  
  
$.when(events_ajax).then(function(results) {  
    ...  
});
```

2. Inside then function, call initDashboard, which is defined in **Dashboard.js** file

```
...    initDashboard(results);
```

## In Dashboard.js

// initMap IF IT DOES NOT EXIST

1. We initialize the map. InitMap function is stored in **map.js** file

```
if (!map) {  
    initMap('map');  
}
```

//CHECK THAT WE HAVE COORDINATES AND ADAPT NEEDED ATTRIBUTES

1. Add an if statement to check coordinates

```
if (d.latitude && d.longitude) {  
    .  
}
```

2. Adapt event\_type, admin1 and source attributes

```
if (!d.event_type || d.event_type === '') {  
    d.event_type = 'Not available';  
}  
if (!d.admin1 || d.admin1 === '') {  
    d.admin1 = 'Not available';  
}  
if (!d.source || d.source === '') {  
    d.source = 'Not available';  
}
```

3. Adapt date format, which comes like '00YY-mm-dd'

```
var dateFormat = d3.time.format('%Y-%m-%d'); // Format in which data come  
var event_date_split = d.event_date.split('-');  
d.event_date = (new Number(event_date_split[0]) + 2000) + '-' + event_date_split[1] + '-' + event_date_split[2];  
d.dd = dateFormat.parse(d.event_date);
```

4. Add dimensions and groups for each desired attribute

```
cf.dim.admin1 = cf.dimension(function(d) { return d.admin1; });  
cf.grp.admin1 = cf.dim.admin1.group().reduceSum(function(d) { return d.fatalities; });
```

5. Initialize charts

```
charts.admin1 = dc.rowChart('#chart-1');
```

## 6. Add charts

```
$(dom.chart1).html('<h4>1st admin level (Top 10)</h4>');
charts.admin1
  .width(250)
  .height(calculateChartHeight(cf.grp.admin1.size(),10, 25, 40))//
  .dimension(cf.dim.admin1)
  .group(cf.grp.admin1)
  .elasticX(true)
  .data(function(group) { return group.top(10); })
  .colors(['#6DDC7A'])
  .colorAccessor(function(d, i){ return 0;})
  .xAxis().ticks(3);//.tickFormat(d3.format("g"));

// Date chart
charts.event_date.width(1000)
  .height(120)
  .margins({top: 0, right: 50, bottom: 20, left: 40})
  .dimension(cf.dim.dd)
  .group(cf.grp.dd_COUNT)
  .centerBar(true)
  .gap(1)
  .x(d3.time.scale().domain([new Date(2016, 0, 1), new Date(2017, 0, 1)]))
  .yAxis().ticks(3);
```

## 7. Group all to count filtered events

```
cf.grp.all = cf.groupAll();
```

## 8. Add data count indicator

```
dc.dataCount("#filter-indicator")
  .dimension(cf)
  .group(cf.grp.all);
```

## 9. Add table

```
charts.Table
  .dimension(cf.dim.event_type)
  // Data table does not use crossfilter group but rather a closure
  // as a grouping function
  .group(function (d) {
    return d.event_type;
  })
  // (_optional_) max number of records to be shown, `default = 25`
  .size(cf.grp.data_id.size())
  // There are several ways to specify the columns; see the data-table documentation.
  // This code demonstrates generating the column header automatically based on the columns.
  .columns([
    function(d) {
      return d.location;
    },
    function(d) {
      return d.admin1;
    },
    function(d) {
      return d.fatalities;
    }
  ])
  // (_optional_) sort using the given field, `default = function(d){return d;}`
  .sortBy(function (d) {
    return d.location;
  })
  // (_optional_) sort order, `default = d3.ascending`
  .order(d3.ascending)
  // (_optional_) custom renderlet to post-process chart using [D3] (http://d3js.org)
  .on('renderlet', function (table) {
    table.selectAll('.dc-table-group').classed('info', true);
  });
```