



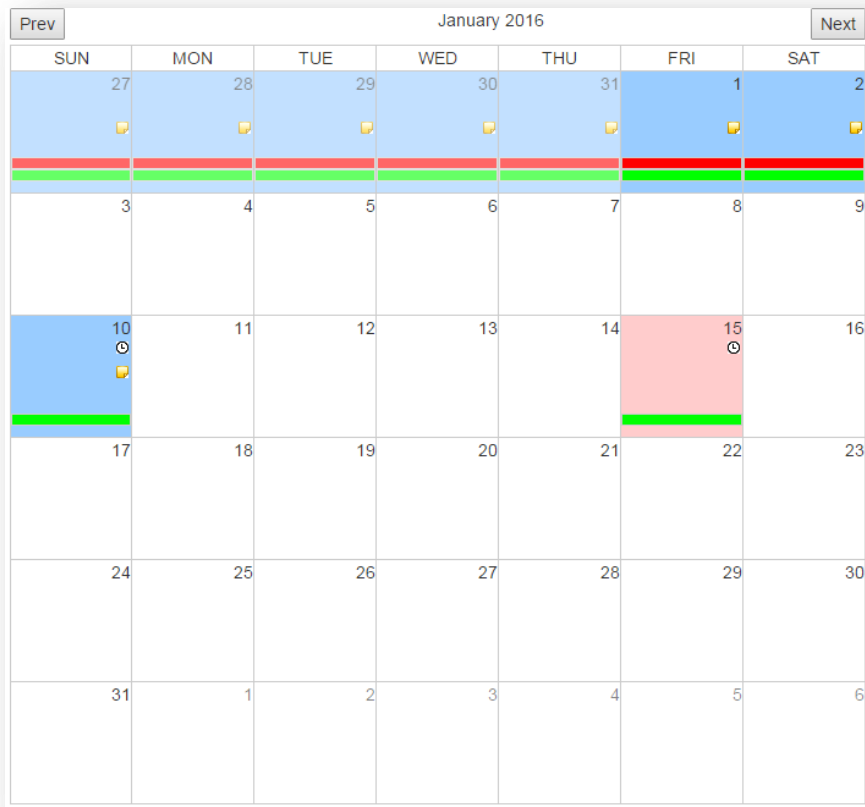
# **JQuery Based Calendar**

**Submitted by: Hunaid Husain**

**Date: 06/01/2016**

## Contents

Usage .....	2
Simple.....	2
Populate Calendar .....	4
Background Colour.....	5
Icons .....	6
Events .....	7
Ajax Support .....	8
Trigger Events.....	9
Prev Button.....	9
Next Button .....	9
Day Click .....	9
Destroy Calendar .....	10
Table size.....	10



## Features:

1. Full Calendar with Prev. and Next buttons.
2. Customize to add events, icons and different background colours with CSS.
3. Events available on click of day and events.

## Usage

### Simple

The simplest usage of Calendar is to display the calendar with just empty dates.

Prev		March 2016					Next	
SUN	MON	TUE	WED	THU	FRI	SAT		
28	29	1	2	3	4	5		
6	7	8	9	10	11	12		
13	14	15	16	17	18	19		
20	21	22	23	24	25	26		
27	28	29	30	31	1	2		
3	4	5	6	7	8	9		

To configure this on your HTML page to below steps:

1. Configure dependencies

In the head tag, place below js and css files. Make sure they are present in the folders specified.

```
<head>
  <link href="css/calendar.css" rel="stylesheet">
  <script src="js/jquery.min.js"></script>
  <script src="js/moment.js"></script>
  <script src="js/calendar.js"></script>
  <link href="css/bootstrap.min.css" rel="stylesheet">
</head>
```

2. Initialize the Calendar

In the script tag, place the below code snippet.

```
<script>
  var jsonData=null;
  var currentDate=null;
  $( document ).ready(function(){
    // call createCalendar with null data. This will create a empty calendar with
    only dates.
    $(".calendar").createCalendar(jsonData);
  });
</script>
```

Here 2 variables are defined; both are not assigned any value. If you want to open the calendar on a particular month and year. Populate the **currentDate** variable.

```
var currentDate='06-03-2016';
```

### 3. Position the calendar

Place the below div, where you want the calendar to appear.

```
<div class="calendar"></div>
```

That's it. You should see the calendar with current month.

## Populate Calendar

February 2016						
Prev	SUN	MON	TUE	WED	THU	FRI
	31	1	2	3	4	5
	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	1	2	3	4
	6	7	8	9	10	11

The Calendar date can be populated with background colour, events and icons. All these data can be provided to the calendar from a JSON object.

Let's first consider the background colour.

## Background Colour

The JSON String you need to provide is

```
{
  "jsondata": [
    {
      "date": "27-12-2015",
      "datetype": "busy"
    }
  ]
}
```

So now the Initialisation of calendar script becomes.

```
var jsonData=null;
var currentDate=null;

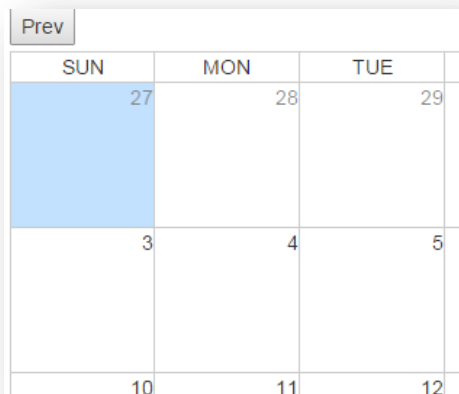
$( document ).ready(function(){
  var jsonString='{"jsondata":[{"date":"27-12-2015","datetype":"busy"}]';
  jsonData = jQuery.parseJSON(jsonDataString)
  $(".calendar").createCalendar(jsonData,currentDate);
});
```

After this we also need to add a css class for the colour.

```
.calendarTable td.busy
{
  background-color: #99ccff;
}
```

Note here that the *busy* in the json object matches the css class.

That's it. You will get your day coloured.

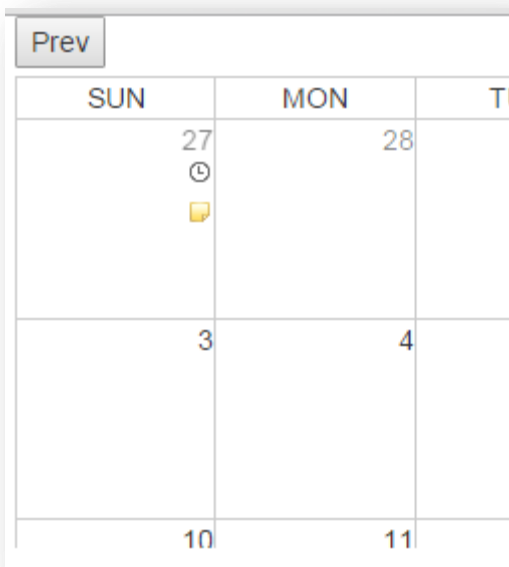


Prev	SUN	MON	TUE	
	27	28	29	
	3	4	5	
	10	11	12	

If you want 2 days to be of same color. Use this JSON data.

```
{
  "jsondata": [
    {
      "date": "27-12-2015",
      "datatype": "busy "
    },
    {
      "date": "31-12-2015",
      "datatype": "busy "
    }
  ]
}
```

## Icons



You can show icons on the day as well. Add `iconlist` attribute in the `jsondata`.

```
{
  "jsondata": [
    {
      "date": "27-12-2015",
      "iconlist": "appointment, notes"
    }
  ]
}
```

Multiple icons can be provided with a comma separated list.

In the css provide below classes

```
div.appointment{
```

```

    top: 15px;
    right: 2px;
}

```

Above style will position the icon on the day.

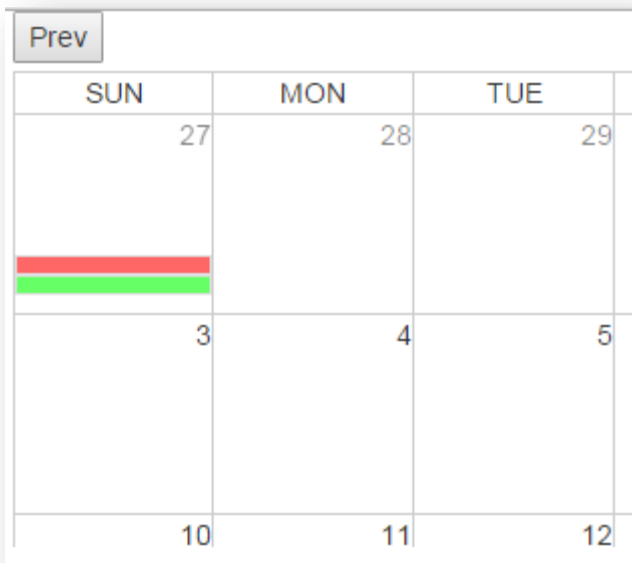
```

img.appointment{
content:url("appointment.png");
}

```

Above style specifies the icon image path

## Events



Prev	SUN	MON	TUE
	27	28	29
	3	4	5
	10	11	12

Events can be added to the calendar using JSONdata.

```

{
  "jsondata": [
    {
      "date": "27-12-2015",
      "events": [
        {
          "eventtype": "reminder",
          "eventtext": "reminder text"
        },
        {
          "eventtype": "appointment",
          "eventtext": "reminder text"
        }
      ]
    }
  ]
}

```



```

    }
  ]
}
]
}

```

In the CSS add below classes

```

div.reminderevent{
    border:1px solid #ccc;
    background-color: #ff0000;
    top: 70px;
}

```

```

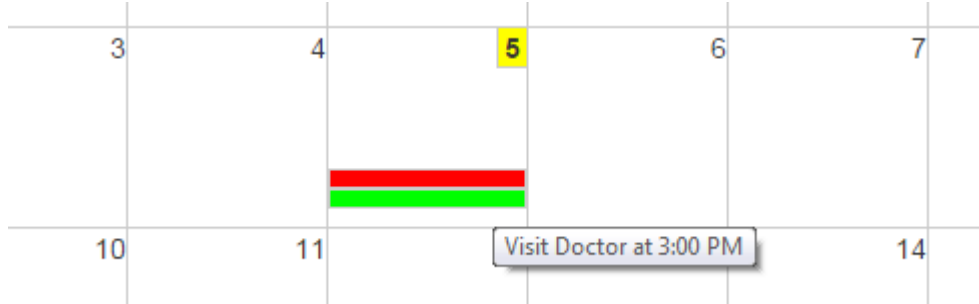
div.appointmentevent{
    border:1px solid #ccc;
    background-color: #00ff00;
    top: 80px;
}

```

Note the extra *event* added to the class name as mentioned in the JSONdata.

The above style provides the styling of the event and its position.

The `eventtext` attribute is used as hover text on the event as shown below.



## Ajax Support

The JSONdata can also be populated using ajax REST calls.

Make following changes to script to initialize the calendar.

```

var jsonData=null;
var currentDate;

$( document ).ready(function(){

```

```

$.ajax({
  url: "http://localhost:8080/json/jan-2016",
  dataType: 'json',
  async: false,
  success: function(data) {
    jsonData=data;

    $(".calendar").createCalendar(jsonData);
  }
});

```

## Trigger Events

On clicking on the Next and Prev buttons, new JSONData can be passed to the calendar. Implement the below methods.

### Prev Button

```

function beforePrev(){

$.ajax({
  url: "http://localhost:8080/json/dec-2015",
  dataType: 'json',
  async: false,
  success: function(data) {
    jsonData=data;
  }
});
}

```

### Next Button

```

function beforeNext(){
$.ajax({
  url: "http://localhost:8080/json/feb-2016",
  dataType: 'json',
  async: false,
  success: function(data) {
    jsonData=data;
  }
});
}

```

### Day Click

Click event on the day can be intercepted by implementing method as below

```

function dayClickedEvent(dayClicked){

}

```

Here the dayClicked is the string specifying the date (01-05-2015)

## Destroy Calendar

Call this method to clear the calendar content.

```
$(".calendar").empty();
```

## Table size

The Calendar is responsive. The initial size of the calendar can be provided by below style class. Mention this class at the bottom of your style tag. The default size is 100px;

```
.calendarTable td.size {  
    height: 70px;  
    width: 70px;  
}
```

January 2016						
Prev						Next
SUN	MON	TUE	WED	THU	FRI	SAT
27 📅	28 📅	29 📅	30 📅	31 📅	1 📅	2 📅
3 📅	4 📅	5 📅	6 📅	7 📅	8 📅	9 📅
10 📅 🕒	11	12	13	14	15 🕒	16
17 📅	18	19	20	21	22 📅	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6