Getting Started with Timeline

From SIMILE Widgets

Contents

Getting Started

Here are a few easy steps to create a simple timeline. Open up your favorite text or HTML editor and start creating an HTML file.

Note

This tutorial has been partially updated for use with Timeline version 2.x. The screen shots are from Timeline version 1, so your Timeline will look somewhat different.

Examples

In addition to this tutorial, please check out:

- The local Timeline example: (http://simile-widgets.googlecode.com/files/timeline_local_example_1.0.zip) a Timeline web page and data file that doesn't require a web server to use.
- Timeline examples (http://simile-widgets.googlecode.com/svn/timeline/trunk/src/webapp/site/examples/)

Step 1. Link to the API

In your HTML code, link to Timeline's Javascript API code as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />
    ...
    <script src="http://api.simile-widgets.org/timeline/2.3.1/timeline-api.js?bundle=true" type="text/ja
    ...
  </head>
  <body>
```

```
     ...
   </body>
 </html>
```

Step 2. Create a DIV Element

Create a ***div*** element in your HTML code, and include a noscript element immediately after it.

```
<div id="my-timeline" style="height: 150px; border: 1px solid #aaa"></div>
<noscript>
This page uses Javascript to show you a Timeline. Please enable Javascript in your browser to see the ful
</noscript>
```

You should give it an ID as well as fix its height. You can optionally set its borders, this usually makes the timeline look better.

The noscript tag will help out people who have turned off Javascript in their browser. Timeline uses Javascript, which is included in all browsers and enabled by default. It does not use Java.

Step 3. Call *Timeline.create()*

Add two event handlers, ***onload*** and ***onresize***, to the ***body*** element:

```
  <body onload="onLoad();" onresize="onResize();">
    ...
  </body>
```

Then write the following code in a ***script*** block or a separate Javascript file:

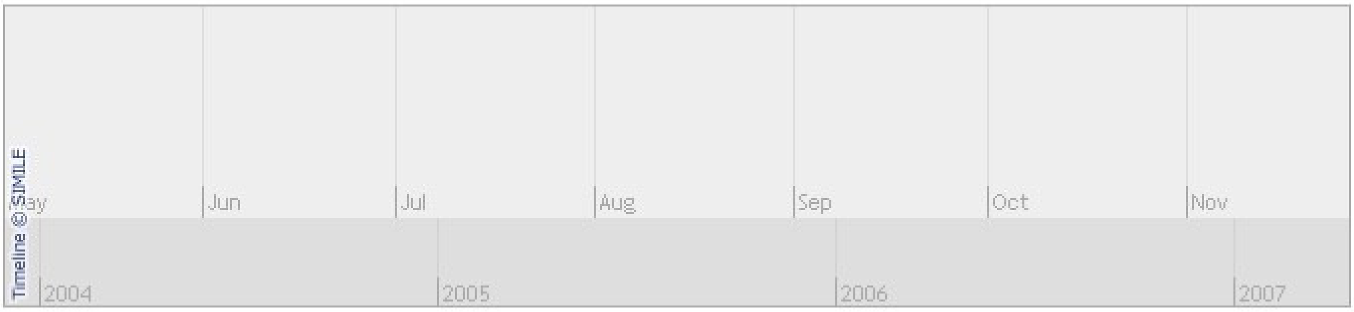
<source lang="javascript" lines=0>

```
var tl;
function onLoad() {
  var bandInfos = [
    Timeline.createBandInfo({
        width:          "70%",
        intervalUnit:   Timeline.DateTime.MONTH,
        intervalPixels: 100
    }),
    Timeline.createBandInfo({
        width:          "30%",
        intervalUnit:   Timeline.DateTime.YEAR,
        intervalPixels: 200
    })
  ];
  tl = Timeline.create(document.getElementById("my-timeline"), bandInfos);
}
```

```
var resizeTimerID = null;
function onResize() {
    if (resizeTimerID == null) {
        resizeTimerID = window.setTimeout(function() {
            resizeTimerID = null;
            tl.layout();
        }, 500);
    }
}
```

</source>

Note that we put the code to construct the timeline in the ***onload*** handler to ensure that when we start to use Timeline's API, all its code has been loaded. That code creates a horizontal timeline (below) with 2 bands: in the top band, a month spans 100 pixels (approximately, since a month here refers to 30 days while not every month is exactly 30 days long); and in the bottom band, a year spans 200 pixels. The top band takes up 70% of the timeline's height, and the bottom band 30%. **Note that the two bands scroll independently.**



Step 4. Keep the bands in sync

To make the two bands scroll in synchrony, and then to make the bottom band highlights the visible time span of the top band, add the following code:

<source lang="javascript" lines=0>

```
  bandInfos[1].syncWith = 0;
  bandInfos[1].highlight = true;
```

</source>

Gives you this...

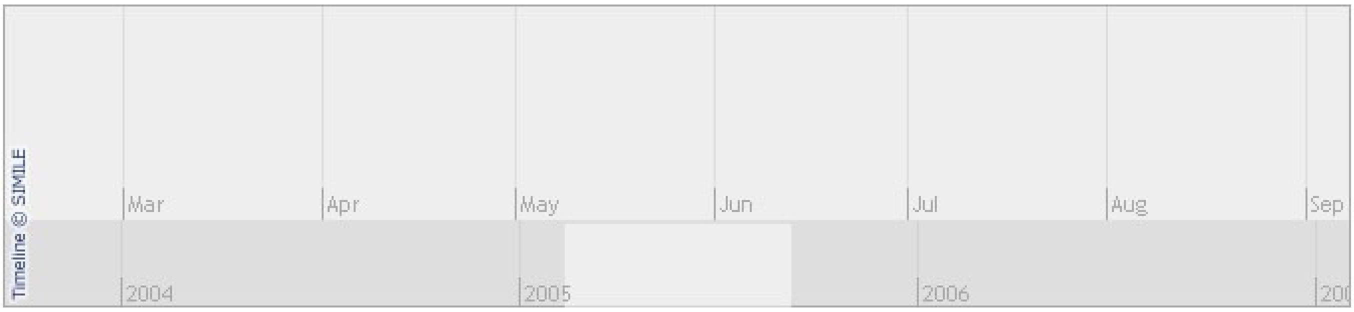
<source lang="javascript" lines=0>

```
function onLoad() {
  var bandInfos = [
    Timeline.createBandInfo({
        width:          "70%",
        intervalUnit:   Timeline.DateTime.MONTH,
        intervalPixels: 100
    }),
    Timeline.createBandInfo({
        width:          "30%",
        intervalUnit:   Timeline.DateTime.YEAR,
        intervalPixels: 200
    })
  ];
  bandInfos[1].syncWith = 0;
  bandInfos[1].highlight = true;

  tl = Timeline.create(document.getElementById("my-timeline"), bandInfos);
}
```

</source>

If you try to pan one band, the other is scrolled as well.



Step 5. Add Events

To add events to the timeline, create a !DefaultEventSource as shown below. Then load the event source with data from your XML, JSON or SPARCL event file. See Event attributes and loading event files. It is not hard for developers to add additional loaders for other event file formats. Additional information on event source. Add the following code:

<source lang="javascript" lines=0>

```
  ...
  var eventSource = new Timeline.DefaultEventSource();
  ...
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
  ...
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
  ...
  Timeline.loadXML("example1.xml", function(xml, url) { eventSource.loadXML(xml, url); })
```

</source>

Gives you this...

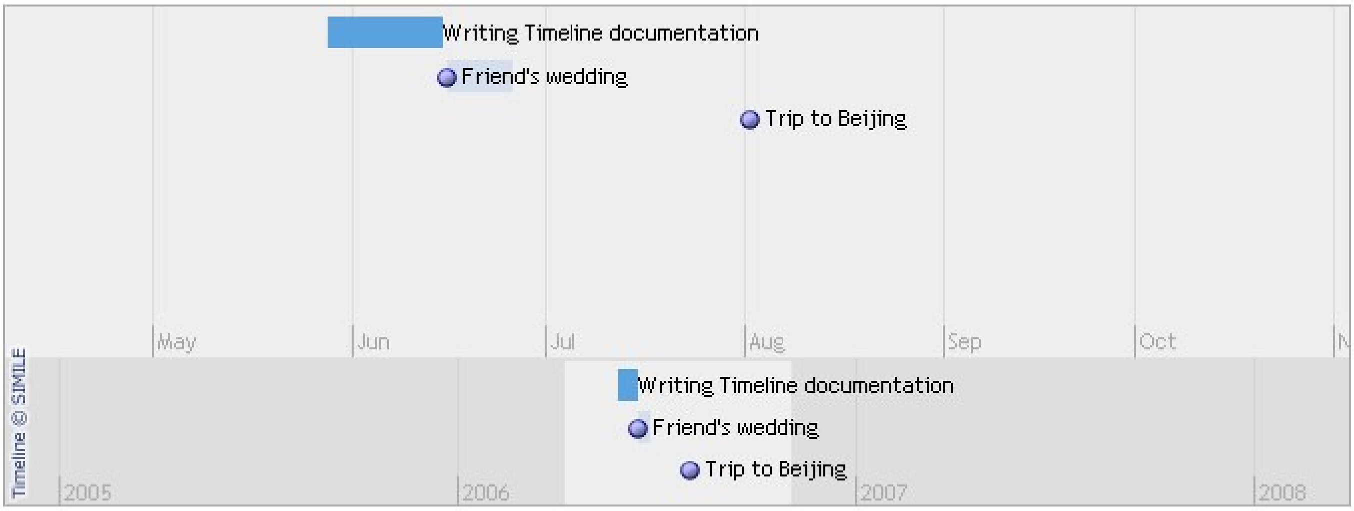
<source lang="javascript" lines=0>

```
function onLoad() {
  var eventSource = new Timeline.DefaultEventSource();
  var bandInfos = [
    Timeline.createBandInfo({
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
        width:          "70%",
        intervalUnit:   Timeline.DateTime.MONTH,
        intervalPixels: 100
    }),
    Timeline.createBandInfo({
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
        width:          "30%",
        intervalUnit:   Timeline.DateTime.YEAR,
        intervalPixels: 200
    })
  ];
  bandInfos[1].syncWith = 0;
  bandInfos[1].highlight = true;

  tl = Timeline.create(document.getElementById("my-timeline"), bandInfos);
  Timeline.loadXML("example1.xml", function(xml, url) { eventSource.loadXML(xml, url); });
}
```

</source>

The date field parm was added to make sure the timeline starts out showing the events immediately without requiring the user to pan first. If you do not provide a date parm, the default is ***now***. Here is the resulting timeline with 3 events:



Take a look at example1.xml. There are 3 types of events:

1. a duration
2. an instantaneous event with an imprecise starting time
3. an instantaneous event with a precise starting time

Click on the events to see how their bubbles are rendered based on the data in the XML file. For the exact format of such XML files, refer to the documentation on event sources. **Note that loading XML files is only one way in which you can add events to timelines.**

Step 6. Differentiate the two bands

Looking at the previous timeline, it is obvious that the lower band looks denser, and it will become a lot denser a lot quicker than the upper band should we add more events. Usually, a lower band usually acts as a zoomed-out overview for an upper band and it does not have to show as much detail as the upper band. Change the lower band to be an overview band:

<source lang="javascript" lines=0>

```
 ...
     overview:       true,
```

</source>

Gives you this...

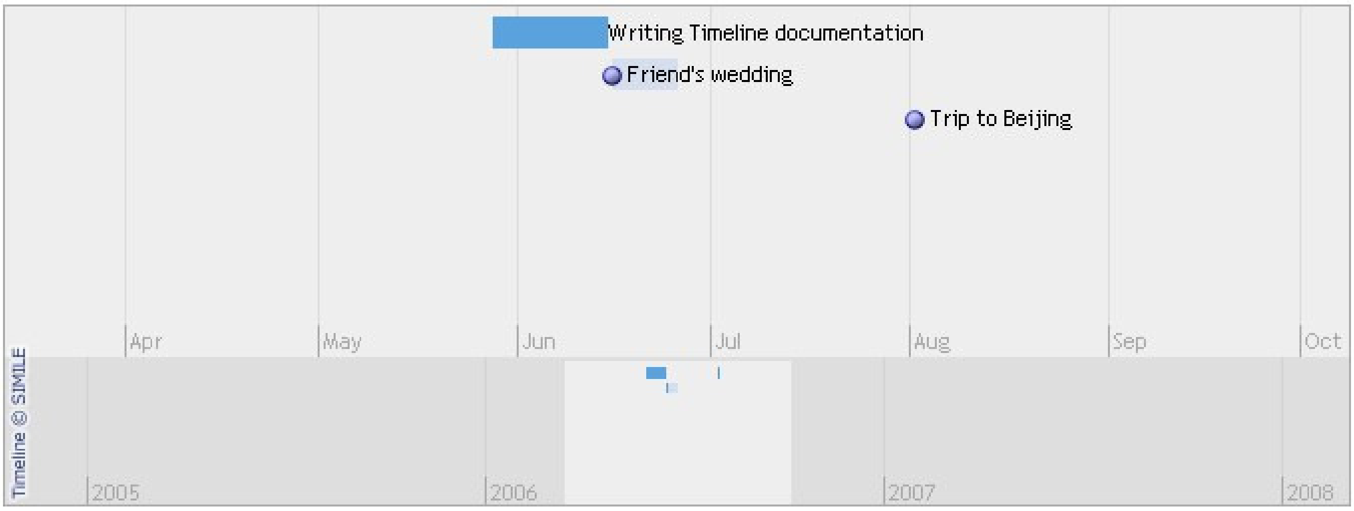
<source lang="javascript" lines=0>

```
function onLoad() {
  var eventSource = new Timeline.DefaultEventSource();
  var bandInfos = [
    Timeline.createBandInfo({
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
        width:          "70%",
        intervalUnit:   Timeline.DateTime.MONTH,
        intervalPixels: 100
    }),
    Timeline.createBandInfo({
        overview:       true,
        eventSource:    eventSource,
        date:           "Jun 28 2006 00:00:00 GMT",
        width:          "30%",
        intervalUnit:   Timeline.DateTime.YEAR,
        intervalPixels: 200
    })
  ];
  bandInfos[1].syncWith = 0;
  bandInfos[1].highlight = true;

  tl = Timeline.create(document.getElementById("my-timeline"), bandInfos);
  Timeline.loadXML("example1.xml", function(xml, url) { eventSource.loadXML(xml, url); });
}
```

</source>

The lower band of the timeline below does not show text and its event markers are also smaller.



For more background on how a timeline is initialized including how to override defaults check out Understanding Initialization

That's it for getting started. To continue with this tutorial check out creating Hot Zones when your events get too cramped.

If you have questions, please use the mailing list (http://groups.google.com/group/simile-widgets/) . Comments posted here are NOT read

Retrieved from "http://www.simile-widgets.org/w/index.php?title=Getting_Started_with_Timeline&oldid=773"

Categories: Timeline | Getting Started

- This page was last modified on 22 November 2010, at 18:27.
- This page has been accessed 23,945 times.
- Content is available under Attribution 3.0 Unported.