

Time Zone Chart Specification

Overview

The Time Zone Chart will be a single-page web application allowing a user to create a chart to visually show what time zones they've been in. The chart will show UTC time on the x-axis and UTC offset on the Y axis, and will indicate the locations the user is at for any given UTC time.

Time Zone Changes: March & April 2016

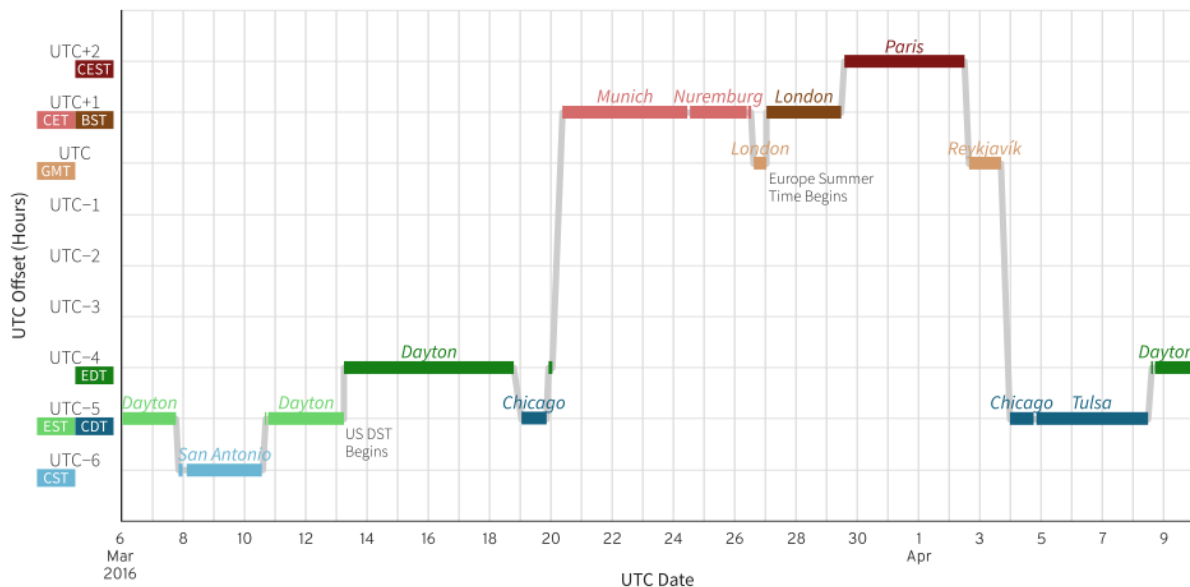


Figure 1: Example Time Zone Chart

Physical Architecture

Time Zone Chart will exist as part of Paul Bogard's Portfolio; as such, it will be hosted on AWS via Heroku.

Software Architecture

Since Time Zone Chart will be part of Paul Bogard's Portfolio, it will technically be written as part of a Ruby application on the Rails framework. However, as this will be a single page application, the functionality unique to the Time Zone Chart will be written in JavaScript using the jQuery library.

This application will not have persistent storage, so no database will be used. However, the user shall have the ability to generate a link which encodes the parameters in the querystring, so that they can come back to or share a chart.

All times shall be stored as Unix timestamps (integer milliseconds since the Unix Epoch) and converted to strings as necessary for output.

Chart

The chart will be in SVG format and will be generated by JavaScript.

Naming Conventions

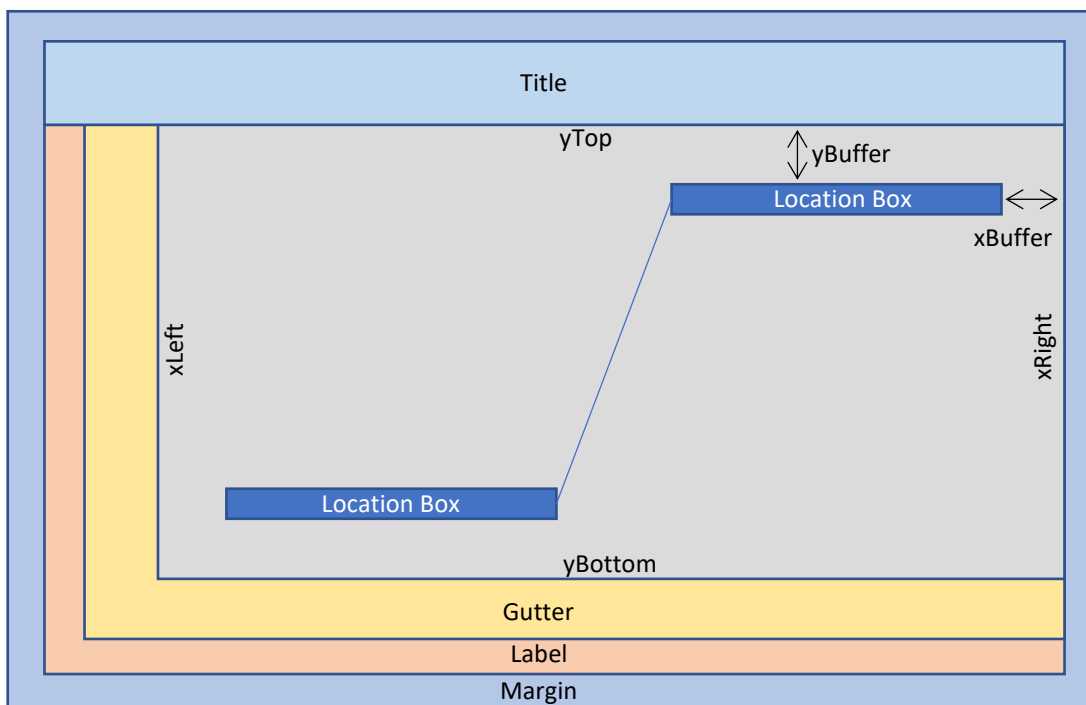
For this specification, “input” includes both HTML input and HTML select fields unless otherwise specified.

Input fields shall have a CSS class prefixed by `field-` (for example, `field-start` or `field-location`). The table data cell containing each field shall have a CSS class prefixed by `cell-` (for example, `cell-start` or `cell-location`).

The title field shall have an id and class of `field-title`, and the chart size field shall have an id and class of `field-size`. However, input fields inside rows do not require an id attribute, as they will be accessed by row with jQuery’s `eq()` and `find()` functions.

Insert Row buttons shall have the CSS class `button-insert`, and Delete Row buttons shall have the class `button-delete`.

Table rows that contain input fields shall have the CSS class `row-location`. Table rows that contain buttons but do not contain any input fields shall *not* have the `row-location` class.



Security

Time Zone Chart shall use SSL; any visitors attempting to access the site via HTTP shall be redirected to HTTPS.

User Interaction

Form

Users will be presented with a table allowing them to enter a series of locations:

	UTC Start	Location	Offset	UTC End	
Add		[Location]	[Offset ▼]	[UTC End]	Del
Add	[UTC Start]	[Location]	[Offset ▼]	[UTC End]	Del
Add	[UTC Start]	[Location]	[Offset ▼]	[UTC End]	Del
Add	[UTC Start]	[Location]	[Offset ▼]		Del
Add					

Clicking the add location button shall insert another row at that point, and sensible defaults shall be picked for the UTC start and end times.

Clicking a row's delete button shall delete that row. The table shall always have at least two rows, so the delete button shall not be available if the table has only two rows.

When a form value changes, the chart shall be updated automatically.

The window's query string parameters shall be automatically updated as the form is updated, so that the user can bookmark or share their current state at any time.

Chart

The chart will be automatically generated as the form is updated.

The chart's minimum and maximum on both axes shall be automatically adjusted to fit all the form data with an appropriate margin.

If a user hovers over or clicks on a location box, a tooltip shall display which shows the location name and the arrival and departure times for that location in both that location's local time and UTC. (If the location is in UTC, then only one arrival and departure time shall be displayed.)

Activating any tooltip shall hide any other tooltips visible.

The user shall have the ability to save the current chart as an SVG file.

The user shall have the ability to show the chart full screen (which shall hide the location entry table). The full screen view shall have a link to close the full screen view and return to the normal edit view. All location data shall be maintained while switching between views.