

This document provides a guide to using an MS Excel based workbook to generate a **Json file** for use with the Knightlab TimelineJS3 project.

In developing the workbook, the objective was to simplify the creation of a Json file and support a local (localhost) development setup, thereby negating the requirement for 'Google Sheets'. The notes below are user instructions/guides to using the workbook once opened in Excel.

Setting up a Local server (localhost) and using Joomla Contact Management System (CMS) are out with the scope of this project, however some notes are provided in an attached PDF for those with a technical background. The notes may be of help to users in setting up the call to Knightlab TimelineJS3 software using the Json file produced by the Workbook.

The worksheet is fully open including access to the Visual Basic for Applications (VBA) code, anyone wishing to use in a non-technical application may wish to protect/hide some of the features. (As always take backups)

The Knightlab TimelineJS3 documentation should be read in conjunction with these notes in relation to cell content, requirements etc.

The routine was developed using Excel 2010 and should work with future versions of Excel that support the use of VBA (i.e. NOT the web based versions). Prior to loading the workbook, setting the Excel option to display cells with a numeric zero value as blank can reduce clutter.

On opening the workbook, the user will be presented with 4 worksheets, the use and function of each follows:

### **params**

There are only 2 parameters.

In cell B9 the name and the location for the created/updated Json file should be given. Examples:

Windows Desktop: C:\Users\XXXXXXXX\Desktop\Develop.json

Where 'XXXXXXXX' is user name, 'Develop.json' is the filename (any valid, relevant name with Json Suffix)

Localhost web folder: C:\wamp64\www\XXXXXXXX\timeline3\Develop.json

Where 'XXXXXXXX' is the website name, 'timeline3' is the folder from Knightlab with JavaScript, CSS etc., 'Develop.json' is filename (any valid, relevant name with Json Suffix)

The value in cell B4 is used to restrict the number of rows (slides) in the project, not limited but a higher number will result in small increase in processing time. (Knightlab suggest a timeline with 20 – 30 slides)

Other cells on this worksheet will be updated when the Json file is created / updated.

### **project**

The button in top left cell (A1) will create/update the Json file and can be selected at any point. It can be used on an empty Worksheet to verify file name and location. Note: Knightlab TimelineJS3 will give an error if no events are entered!

This worksheet contains all of the data required to create the Json file. In simple terms, it replaces the Google Sheets Template. Whilst the layout is similar to the Knightlab Google Sheets Template, there are additional columns to support all fields which can be used by the TimelineJS3 software.

Columns A to AD should not be changed.

Column AE is used by the routine to facilitate row sorting

All other columns can be used freely, e.g. for Notes, working info etc., only content in columns A to AD is taken into the Json file.

The cells in rows 2 down can be used to enter data as per Knightlab TimelineJS3 guidelines and subsequently changed / updated. All standard functions such as copy /paste individual cells, rows or columns can be used. Excel functions such 'cell Reference' (e.g. =AF9) and 'VLOOKUP' to reference data in other worksheets can also be used.

Rows can be deleted or inserted.

Worksheet 'slide' (see below) can be used to update the rows and is perhaps the easiest method to create a series of events. The individual cells can then be modified as required.

Data from another worksheet can be copied /pasted; alternatively data from a worksheet created by the various data import routines supported by Excel can be mapped to the columns for regular updates to the Timeline. For Example, CSV imported data, ODBC extracts from database sources etc.

The Worksheet can be customised; Cells, Rows or Columns can be coloured, Columns can be hidden to reduce input cells etc. It is recommended that Column A remains visible as this identifies to the user the type of record (row) for which data is being entered. (Note: Freeze Pane option for first column (Col. A) is implemented by default. (optional)

Two Workbooks are supplied:

- a. Default Workbook – All cells Empty
- b. Customised Workbook - Knightlab Official Timeline JS3 Template Customisation (i.e. Excel Workbook customised to resemble GoogleSheets version)

### **slide**

The button to the right of column G is used to insert a slide (row) into the 'project' worksheet. The contents of column B are transposed and inserted as a row. The insert process sorts the rows by record type (see below) and any existing slides (rows) by start date/time. A routine to clean / reformat the presentation is also run.

The button to create/update the Json file needs to be selected for slide to be included in the Json file. If viewing the timeline in a local (localhost) setup a browser refresh is required (F5 or refresh button/symbol) to load the revised timeline

The data is presented vertically as this provides a more concise overview of a single slide. As with cells in 'project' copy/paste techniques can be used, cell colouring etc.

#### Column A:

This is a list of all the data fields supported by Knightlab TimelineJS3. Comments/notes are attached to most cells and are viewed by 'hovering' over the cell. The comments/notes relate to information on the Knightlab TimelineJS3 site where additional detail is available. These can be edited.

#### Column B:

The data for the slide is entered here. Reference should be made to columns D, E, F, and G which represent the fields which can be used for each record type. For Example a 'TITLE' record would have 'Title' in row 1, 'Display Date' in row 15, 'Headline' in row 16, 'Unique Id' in row 30 any data in other rows will be ignored.

Documentation on the Knightlab website provides fuller information on the content which can be entered for each record type.

Each record type is assigned a 'Sort ID' which is shown in column AE of 'project'; this is used to support presentation on 'project' workbook. (Scale-ID 1, Title-ID 2, Era-ID 3, Event-ID 4). Where there are multiple records allowed (Era and/or Events) the 'Start Date/Time' is used to order the records

#### Column C:

The narrative in each cell is for information only, giving guidance on entry formats etc. and the type of data which is supported. Again, fuller information is available on the Knightlab TimelineJS3 site. The content can be changed/updated if required.

#### Columns D / E / F / G:

These columns are a visual representation of the content allowable for each record type and are for information only.

It should be noted that Knightlab TimelineJS3 software specification for record types is as follows:

Scale: Optional – Only 1 per timeline

Title: Optional – Only 1 per timeline

Eras: Optional – Multiple Allowed (Suggest max of 6, CSS created for 6 different colours)

Events: Required – Multiple Allowed (Minimum of 1, Knightlab suggests max of 20 – 30)

(Note: The 'Scale' record will be accepted by the latest version (3.8.26) of the Knightlab TimelineJS3 software but is not used ('Determining scale dynamically: human' is message given))

#### **HTML support**

This workbook is not required to support the 'project' creation of the Json file but has been included as a simple aid to creating HTML strings for inclusion (copy / paste) in slides. In the first example the text in cells B2, B3, B4, B5, B6 is concatenated to the string in cell B8. This can be copied / pasted to the cells in the worksheets 'project' and / or 'slide'.

Cells B11,B12,B13,B14,B15 are concatenated to Cell B17 and offer a slightly different presentation

The ZIP file attached has the following files:

Develop.xlsm	- Project Workbook with no content
Excel Official TimelineJS3.xlsm	- The Knightlab Official Google Sheets Example in Excel (Customised Project Workbook with Content)
Develop.json	- Json version of empty workbook (gives error: No Events)
Excel Official TimelineJS3.Json	- The Json file for the Official Example, Excel generated
Whitney.json	- Excel generated Json file (Created by taking Json file from Knightlab Github source) Converted to Excel by free web JSON to EXCEL converter Columns mapped to Project Workbook New Json file generated) (Content errors consistent with Knightlab Version, some YouTube content no longer available)
Whitney Screenshot.jpg	Screen Shot of from Whitney Timeline
Localhost-Joomla Notes.pdf	Notes on setting up a Local server / Joomla environment for local in-house timeline development
Article.txt	File to create TimelineHS3 options Joomla Article Call
Timeline Options.pdf	PDF version of Article.txt

### **Github Enhancements Addressed**

This project may address the following enhancements as referenced in the Github Issues.

For consideration:

#### **#668 – Directly support CSV configuration**

The CSV files could be imported to EXCEL and mapped to Timeline Project  
For single conversion, Copy / Paste is quick and simple for columns  
For regular creation / update from standard CSV layout a Mapping Macro could be created

#### **#509 – Darker Font Colours**

The Whitney Screen shot included shows some Custom CSS which works on the  
Timemarkers and Navigation bar. The call to the Timeline CSS includes a call to a Custom CSS  
file.

#### **#489 Change Flag colours for different groups**

The code here worked in TimelineJS3 v 3.6.5;  
Implementing in v 3.8.21 has proved a puzzle (Javascript Skills non-existent)  
Using the Excel Project an additional column could be introduced specifying the Flag  
background colour. This would require some additional Javascript to process but would  
preserve the integrity of Google Sheets based time lines as change only exists in Json file

#### **#438 Dark Theme**

The CSS changes necessary to effect a Theme Colour change could be identified and set-up  
in Excel with meaning full names, cells available to input colour info, opacity etc. with Excel  
creating a custom CSS file which is updated similar to the Json file

**#418 Dynamic Event Loading**

Excel can be fully automated to pull data from different sources including multiple database formats and interfaced with web scrape routines. The data is ultimately stored as conventional spread sheet (Rows / Columns) which could be mapped by VBA to the timeline layout. This opens up numerous development options, for example a timeline line of news events built in 'real' time as they occur, or Sports reporting with Pictures inserted as soon as available. (Excel could be set to update the Json at an appropriate time interval or on request.

The Excel Json project provides a basis but is perhaps better considered as a new project !

**#414 EXCEL to Json Conversion feature**

Request does not have much detail but believe this addresses enhancement request

**#321 Support Era Background Colour**

Version 3.8.21 has coding to support 6 different background colours, these can be set through Custom CSS, they are allocated sequentially (0-5).

The notes above #489 are relevant here, a column for ERA background colour could be introduced, and Javascript applied without affecting legacy timelines

There may be more !