

Timelining Wikipedia Historical Articles – Server-side

Pedro Camilo de Godoy Queiroz

1 Introduction

The idea behind this project is to implement the server-side for a website and a mobile app implemented in the previous courses of the specialization that presents historical articles from Wikipedia in a timeline fashion way.

The aim of the app and the website is to help teachers, students and everybody that is interested in history, to better visualize the historical events and their relationship through the time.

The aim of the server is to use NodeJS, Express and MongoDB to store the data used in the mobile app and website.

2 Expected List of Features

This project will be a prototype. It will store data of a timeline of selected Wikipedia articles.

3 Market Survey

Some examples of websites / app that create timelines:

myHistro (<http://www.myhistro.com>)

STORIES DISPLAYED ON MAPS: GEO-LOCATED INTERACTIVE TIMELINES WITH A SOCIAL TWIST

Watch and read thousands of fascinating timelines, or create your own. Complete with text, video and pictures to create a dynamic timeline mashup.

Using myHistro, you can combine maps and timelines seamlessly into one great presentation, convert any public timeline into a personal pdf file, or export it into Google Earth format for offline storage. All completed timelines can be embedded into your blog and websites for maximum exposure.

TimelineJS (<https://timeline.knightlab.com>)

TimelineJS is an open-source tool that enables anyone to build visually rich, interactive timelines.

Beginners can create a timeline using nothing more than a Google spreadsheet. Experts can use their JSON skills to create custom installations, while keeping TimelineJS's core functionality.

Free-timeline.com (<http://free-timeline.com>)

Free-timeline.com lets you create, save online, and share timelines. It's easy to use on desktop, mobile, or tablets, and even allows import (from Excel or Google Sheets CSV files) and embedding into your own blogs or websites. Best of all, it's absolutely free!

Since 2006, over 600K people have used Free-timeline.com, from teachers and students for learning history, to parents and grandparents for family trees, biographies, and genealogy, to professionals for personal planning, career planning, and even project management. Try it out, it's free.

Free-timeline recently got a major upgrade - if you are looking for your old free-timeline.com work, you can find it at Free-Timeline.NET.

TimeGlider (<https://timeglider.com>)

Web-based timeline software for creating and sharing history, project planning and more ...

Create, collaborate, and publish zooming and panning interactive timelines. It's like Google Maps, but for time.

Timetoast (<https://www.timetoast.com>)

Our timeline tool makes it a breeze to make a timeline, from ancient history to the far-off future.

View your creation in both horizontal and list modes. It works on just about any device and of course it's super quick to update online.

Some of our timeline maker features:

- Add events or multi-date timespans.
- Automatically have your events placed on the appropriate timescale.
- Zoom and pan your timeline.
- Add BCE dates, year-only dates as well as dates with a specific month and day.
- Upload images to bring your timelines to life.
- Adapt to any screen size with our fluid timeline design.

Timeline - Diary and Notes (<https://itunes.apple.com/us/app/timeline-diary-and-notes/id1073862895?mt=8>)

Timeline helps you capture and record the important moments in your life. You can create as many timelines as you want. Using different timeline to track different events.

Create a timeline for your dreams and achievement; a timeline for your travel plan; a timeline for your daily milestone; a timeline with your friends...

Not only the past time, you can create future event and enable notifications. Timeline is now becoming your todo list, your wish list.

Timeline Visualizer

View different content and media from multiple sources chronologically on a visual timeline.

Features:

- On This Day events from Wikipedia
- View your calendar events & photos on a visual timeline
- Easily create personal journal/diary entries and see them on a timeline
- YouTube feed
- Twitter feed
- Google Calendar
- Latest news from The Guardian
- Upcoming and past movie timelines
- Support RSS Feeds
- Create custom timelines and share with others
- Create historical timelines for use as learning tool

4 Design and Implementation

4.1 The REST API Specification

The Rest API specification can be seen at the address:

<https://wikintime.docs.stopligh.io/>

4.2 Database Schemas, Design and Structure

Timeline Info Schema:

```
const Schema = require('mongoose').Schema;

const timelineInfoSchema = new Schema(
  {
    name: {
      type: String,
      required: true,
      unique: true,
    },
    lang: {
      type: String,
      required: true,
    },
  },
  {
    timestamps: true,
  }
);

module.exports = mongoose.model('TimelineInfo', timelineInfoSchema);
```

Timeline Data Schema:

```
const Schema = require('mongoose').Schema;

const dateTimeSchema = new {
  year: {
    type: Number,
    required: true,
  },
  month: {
    type: Number,
    min: 1,
    max: 12,
  },
  day: {
    type: Number,
    min: 1,
    max: 31,
  },
  hour: {
    type: Number,
    min: 0,
    max: 59,
  },
  minute: {
    type: Number,
    min: 0,
    max: 59,
  },
  seconds: {
    type: Number,
    min: 0,
    max: 59,
  },
};

const timelineEventSchema = new Schema(
  {
    source: {
      type: String,
      required: true,
    },
    pageTitle: {
      type: String,
      required: true,
    },
    date: {
      start: dateTimeSchema,
      end: dateTimeSchema,
    },
  },
);
```

```

    {
      timestamps: true,
    }
  );

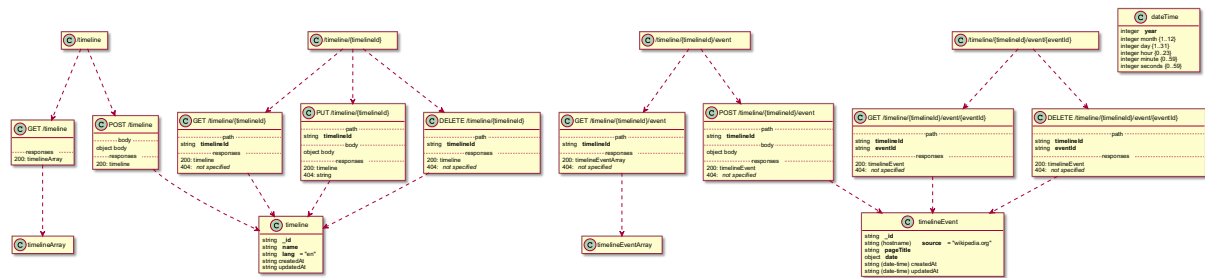
const timelineDataSchema = new Schema(
  {
    timelineId: {
      type: ObjectId,
    },
    data: [timelineEventSchema],
  }
);

module.exports = mongoose.model('TimelineData', timelineDataSchema);

```

4.3 Communication

UML Diagram:



5 Implementation

In order to do not change any code from my previous client implementation, I have decided to change and simplify the Rest API and the database schema.

The Rest API serves now only the GET and POST methods. The POST method adds an event to my unique timeline, and the GET method retrieves all events from the timeline.

The Database Schema and the Communication of the Rest API are shown below.

5.1 Database Schema

```
const mongoose = require('mongoose');

const timelineEventSchema = new mongoose.Schema(
  {
    title: {
      type: String,
      required: true,
    },
    introduction: {
      type: String,
      required: true,
    },
    yearStart: {
      type: String,
      required: true,
    },
    yearEnd: {
      type: String,
      default: '',
    },
    imageUrl: {
      type: String,
      default: '',
    },
    imageCaption: {
      type: String,
      default: '',
    },
    url: {
      type: String,
      required: true,
    }
  },
  {
    timestamps: true,
  },
);

module.exports = mongoose.model('TimelineEvent', timelineEventSchema);
```

5.2.1 POST method

5.2.2 GET method

Replay Import Runner My Workspace + ... No Environment

History

Clear all Save Responses

Todays

- GET http://10.0.1.9:3000/timeline
- GET http://10.0.1.7:3000/timeline
- GET 10.0.1.7:3000/timeline
- GET localhost:3000/timeline
- GET localhost:3000/timeline
- POST localhost:3000/timeline
- GET localhost:3000/timeline
- GET localhost:3000

http://10.0.1.9:3000/timeline

GET http://10.0.1.9:3000/timeline

Params	Authorization	Headers	Body	Pre-request Script	Tests
key					

Body Cookies Headers Tests Test Results

Status: 200 OK Time: 12 ms Size: 21.51 KB Download

```

{
  "yearEnd": "1903",
  "imageId": "https://upload.wikimedia.org/wikipedia/commons/6/61/Eloquen_de_Venezuela_por_las_potencias_europeas_1902.jpg",
  "imageCaption": "Engraving by Willy Stöber depicting the blockade.",
  "url": "https://en.wikipedia.org/wiki/Venezuelan_crisis_of_1902-03",
  "createdAt": "2019-01-15T17:15:12.100Z",
  "updatedAt": "2019-01-15T17:15:12.100Z",
  "title": "Venezuelan crisis of 1902-1903",
  "introduction": "The Venezuelan crisis of 1902-03 was a naval blockade from December 1902 to February 1903 imposed against Venezuela by the United Kingdom, Germany and Italy, over President Cipriano Castro's refusal to pay foreign debts and damages suffered by European citizens in the Venezuelan civil war. Castro assumed that the United States Monroe Doctrine would see the US prevent European intervention, but at the time, US president Theodore Roosevelt and the Department of State saw the doctrine as concerning European seizure of territory, rather than intervention \"per se\". With prior promises that no such seizure would occur, the US allowed the action to go ahead without objection. The blockade saw Venezuela's small navy quickly disabled, but Castro refused to give in, and instead agreed in principle to submit some of the claims to international arbitration, which he had previously rejected. Germany initially objected to this, particularly as it felt some claims should be accepted by Venezuela without arbitration.",
  "yearStart": "1902",
  "url": "https://en.wikipedia.org/wiki/Venezuelan_crisis_of_1902-03",
  "createdAt": "2019-01-15T17:15:12.100Z",
  "updatedAt": "2019-01-15T17:15:12.100Z",
  "title": "Franco-Russian Alliance",
  "introduction": "The Franco-Russian alliance, or Russo-French Rapprochement, was an alliance formed by the agreements of 1891-93; it lasted until 1917. The strengthening of the German Empire, the creation of the Triple Alliance of 1882, and the exacerbation of Franco-German and Russo-German contradictions at the end of the 1880s led to a common foreign policy and mutual strategic military interests between France and Russia. The development of financial ties between the two countries created the economic prerequisites for the Russo-French Alliance.",
  "yearStart": "1894",
  "url": "https://en.wikipedia.org/wiki/Franco-Russian_Alliance",
  "createdAt": "2019-01-15T17:15:12.899Z",
  "updatedAt": "2019-01-15T17:15:12.899Z",
  "title": "Triple Alliance",
  "introduction": "The Triple Alliance was a secret agreement between Germany, Austria-Hungary, and Italy. It was formed on 20 May 1882 and renewed periodically until it expired in 1915 during World War I. Germany and Austria-Hungary had been closely allied since 1879. Italy sought support against France shortly after it lost North African ambitions to the French. Each member promised mutual support in the event of an attack by any other great power. The treaty provided that Germany and Austria-Hungary were to aid Italy if it was attacked by France without provocation. In turn, Italy would assist Germany if attacked by France. In the event of a war between Austria-Hungary and Russia, Italy promised to remain neutral.",
    
```

6 References

The project intends to make use of

- NodeJS (<https://nodejs.org/>)
- Express (<http://expressjs.com/>)
- MongoDB (<https://www.mongodb.com/>)

7 Conclusion

The key difference from this project and the others listed above is the focus to view Wikipedia articles in a timeline way.

Since I have decided to not change the client-side of the application, the implementation of the server-side was pretty straight.

I have decided to do a simpler implementation of the server-side because if the project moves on, I think that the best server-side solution should be to implement a BaaS, like Firebase from Google or Loopback from IBM.