Project

5G00DM06-3006 API Service Development

Sylvi Kokko

Overview

- Inspired by the ItsFactory's real time bus api
- Tampere tram based
- 4 Tables:
  - Tram
  - Stop
  - Line
  - Linestop
- Uses:
  - SQLite
  - Sequelize
  - Node/Express

Technologies

SQLite

- Lightweight, serverless SQL database engine
- Stores data in a single file
- Great for small to medium apps, prototyping, local storage
- Zero configuration

Sequelize

- Promise-based ORM for Node.js
- Supports SQLite, PostgreSQL, MySQL, MSSQL
- Defines models & relationships using JavaScript
- Simplifies database queries
- Makes code easier to read
- Helps prevent SQL injection

Node.js

- JavaScript runtime for network applications
- Enables server-side JavaScript
- Event-driven architecture

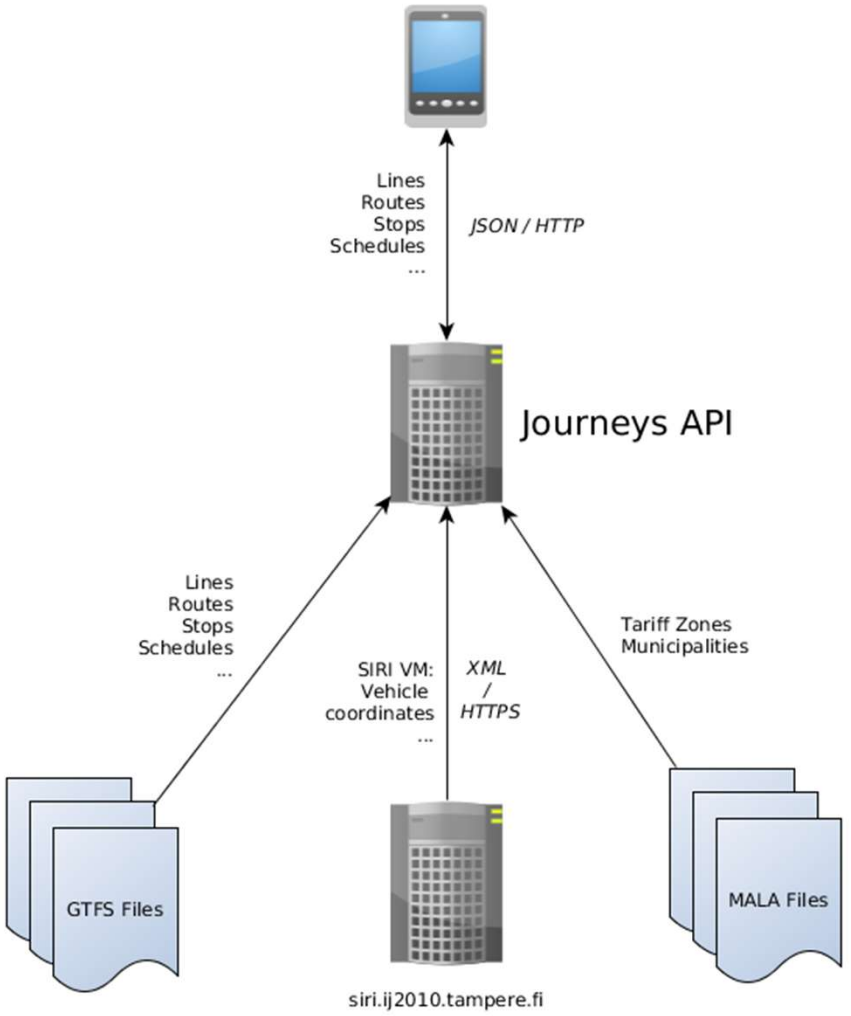
Express.js

- Minimalist web framework for Node.js
- Routing, middleware, request/response handling
- Common foundation for REST APIs & web servers
- Fast API development
- Integration with databases and ORMs like Sequelize

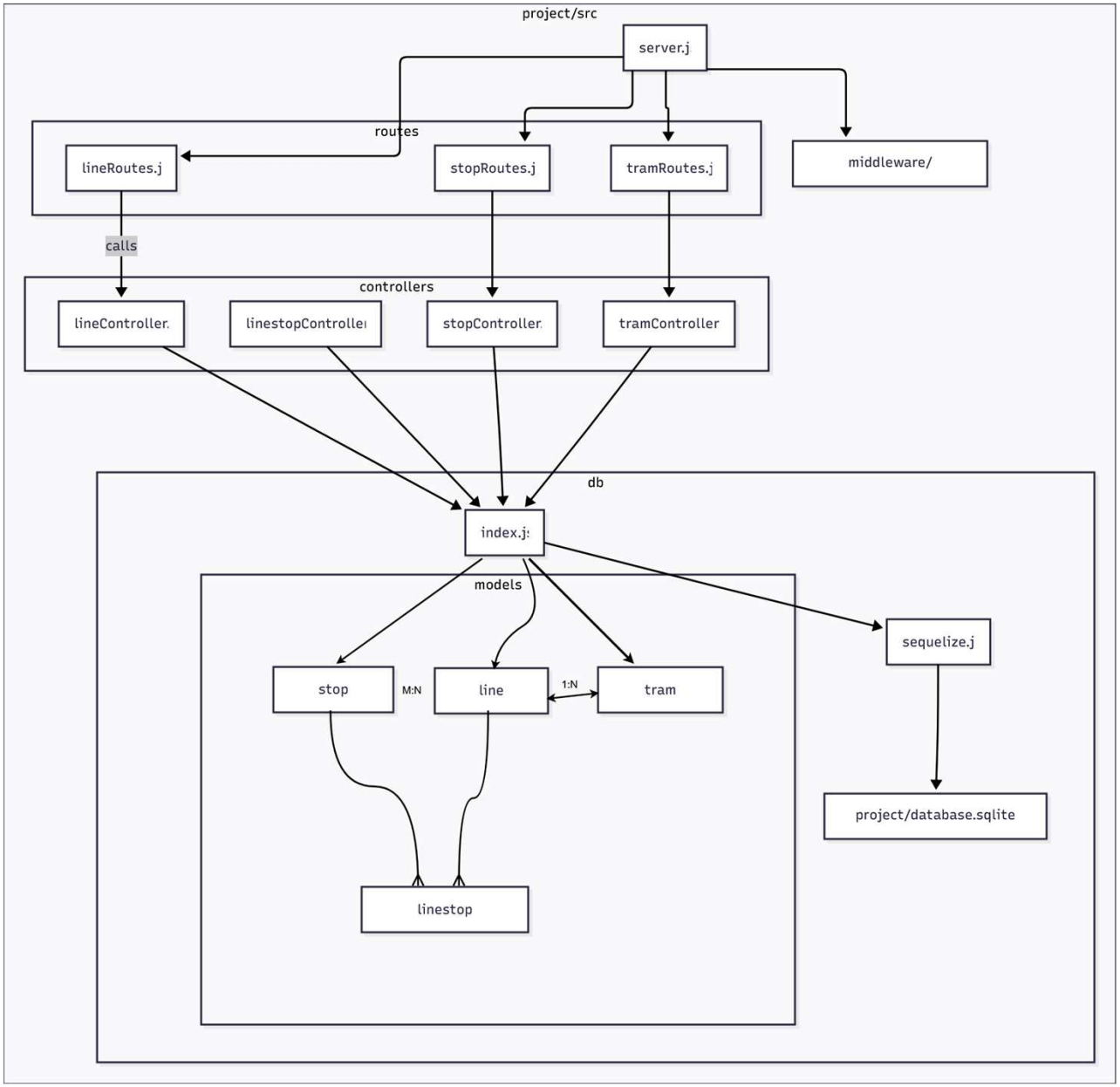
Design

Inspiration

- Itsfactory Journeys API
- Allows developers to access public transport information via REST api
- Routes, lines, journeys, stops, vehicle monitoring
- [Example project from Pasi Kuparinen using the Itsfactory API]

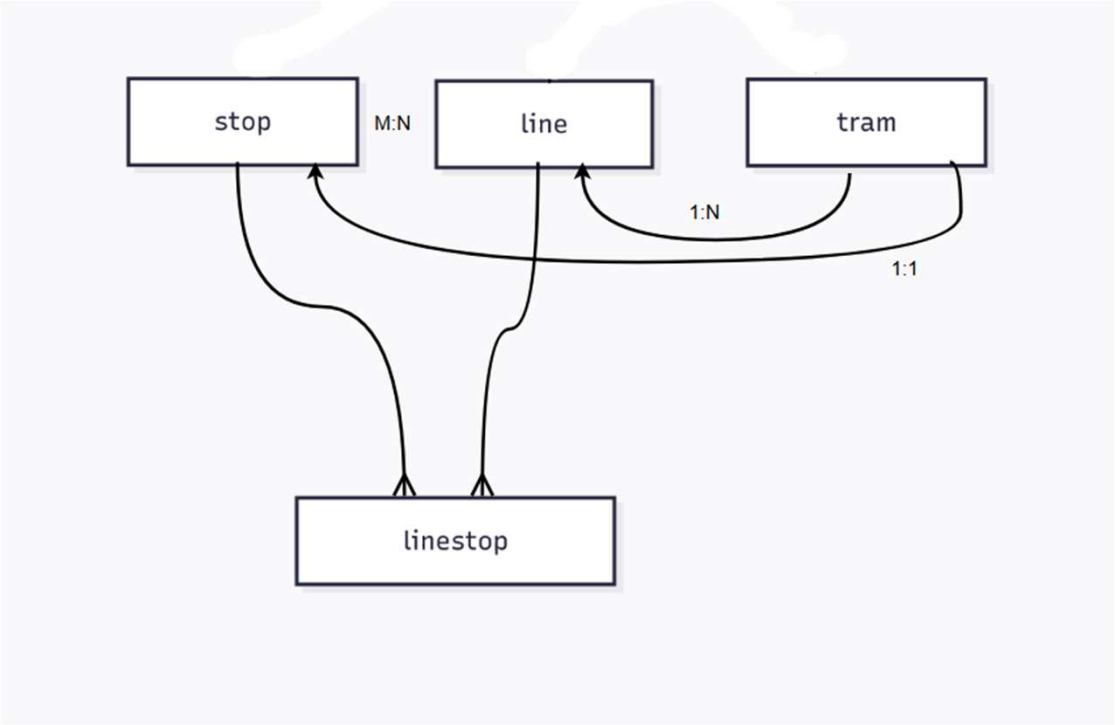


Overall Project Sturcture:



Database

The database uses SQLite and has four tables tram, stop, line and linestop.



Challenges

- How should the entities interact with each other
- Do the entities know what they’re part of
- Should we have a direction entity that knows what side of the road specific stops are on

How itsfactory does it (simplified)

- Buss activity has:
  - origin
  - destination
  - direction
  - onward calls = next stops
    - order

Solutions

- Stop as a table/entity knows nothing about trams or lines
- Line as a table/entity knows nothing about stops
- Linestop as a jointable has a new column called “order” to store the order of the stops for a specific line

Thank you!