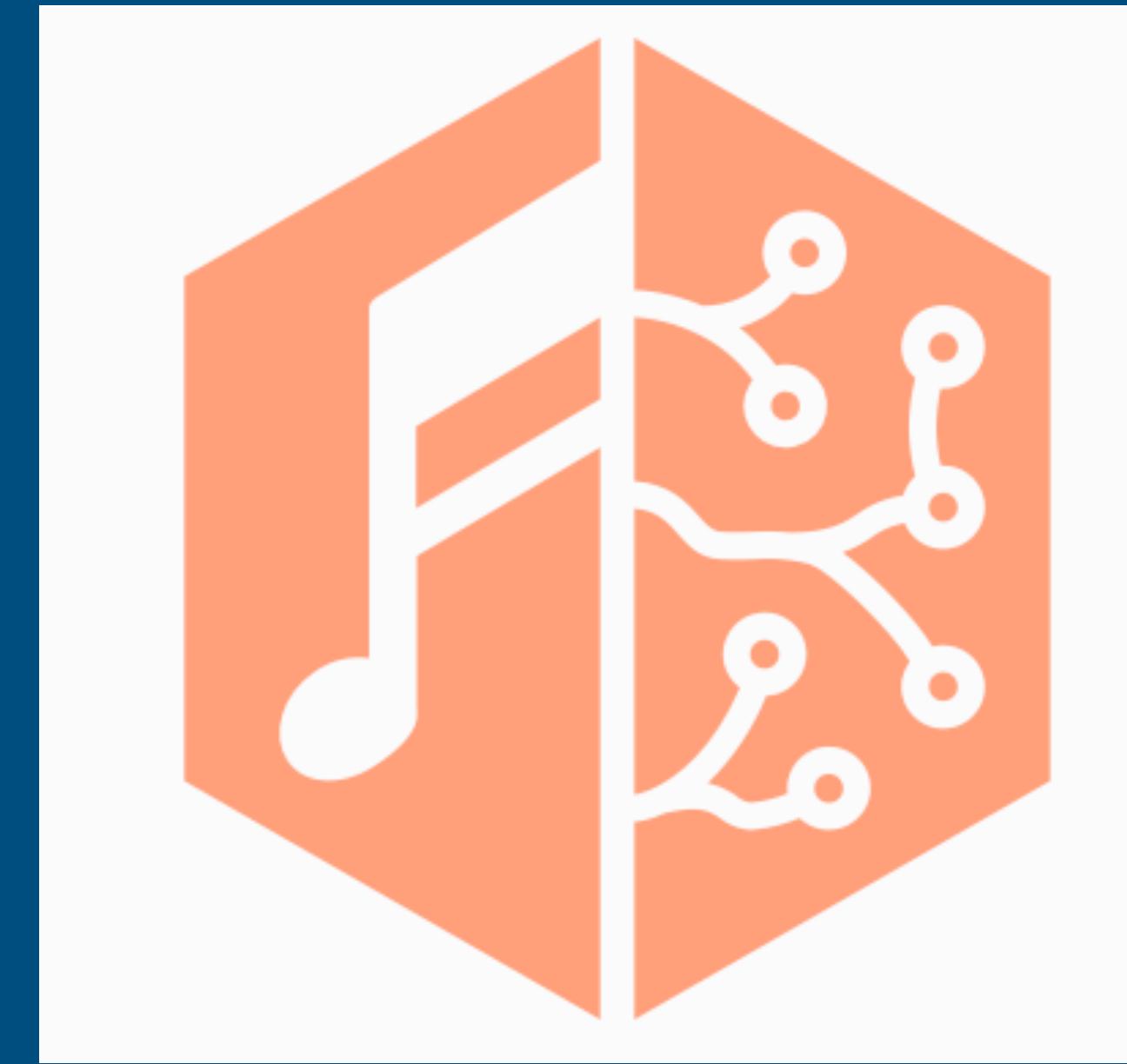


# Playtime Architecture



Full Stack Web Development

# Unified Modelling Language

## RAISING THE LEVEL OF ABSTRACTION



*“Models help us by letting us work at a higher level of abstraction. A model may do this by hiding or masking details, bringing out the big picture, or by focusing on different aspects of the prototype.”*

A screenshot of the official UML website. The background is a dark, abstract digital space with binary code (0s and 1s) and glowing particles. In the center, there's a large, metallic, three-dimensional UML logo. To the left of the logo, the text "INTRODUCTION" is above the large, bold, white title "WHAT IS UML?". Below the title, a subtitle reads "If you're new to modeling and UML®, start here." and a "READ MORE" button is visible. At the bottom of the page, there are several navigation links: "20 YEARS ANNIVERSARY", "WHAT IS UML?", "UML VENDOR", "UML RESOURCES", "UML SPECIFICATIONS", and "OMG UML CERTIFICATION".

INTRODUCTION

# WHAT IS UML?

If you're new to modeling and UML®, start here.

READ MORE

20 YEARS ANNIVERSARY

WHAT IS UML?

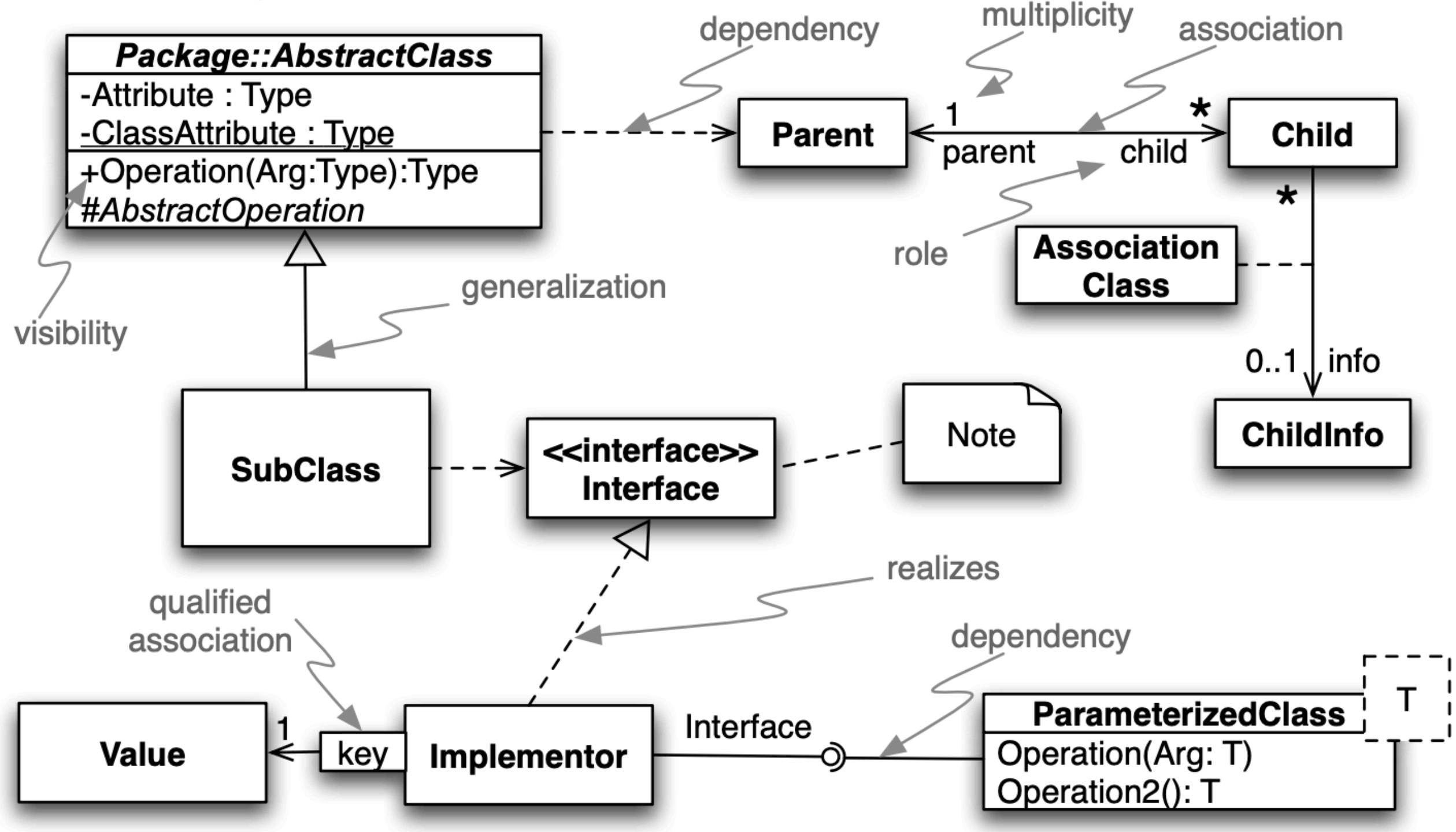
UML VENDOR

UML RESOURCES

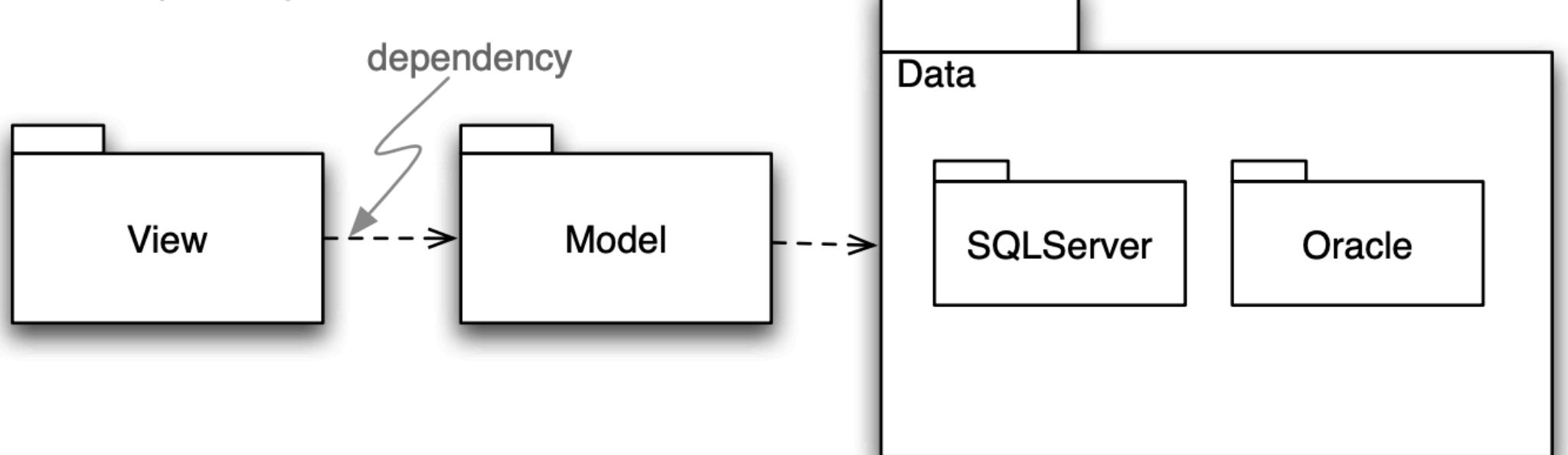
UML SPECIFICATIONS

OMG UML CERTIFICATION

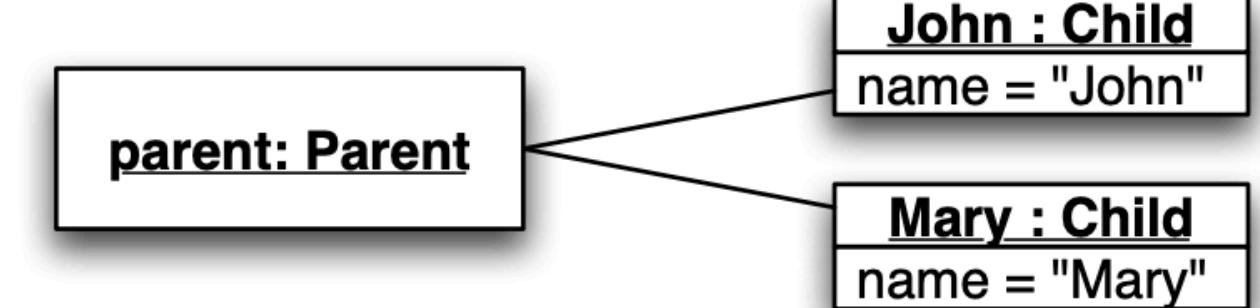
## Class Diagram Elements



## Package Diagram Elements

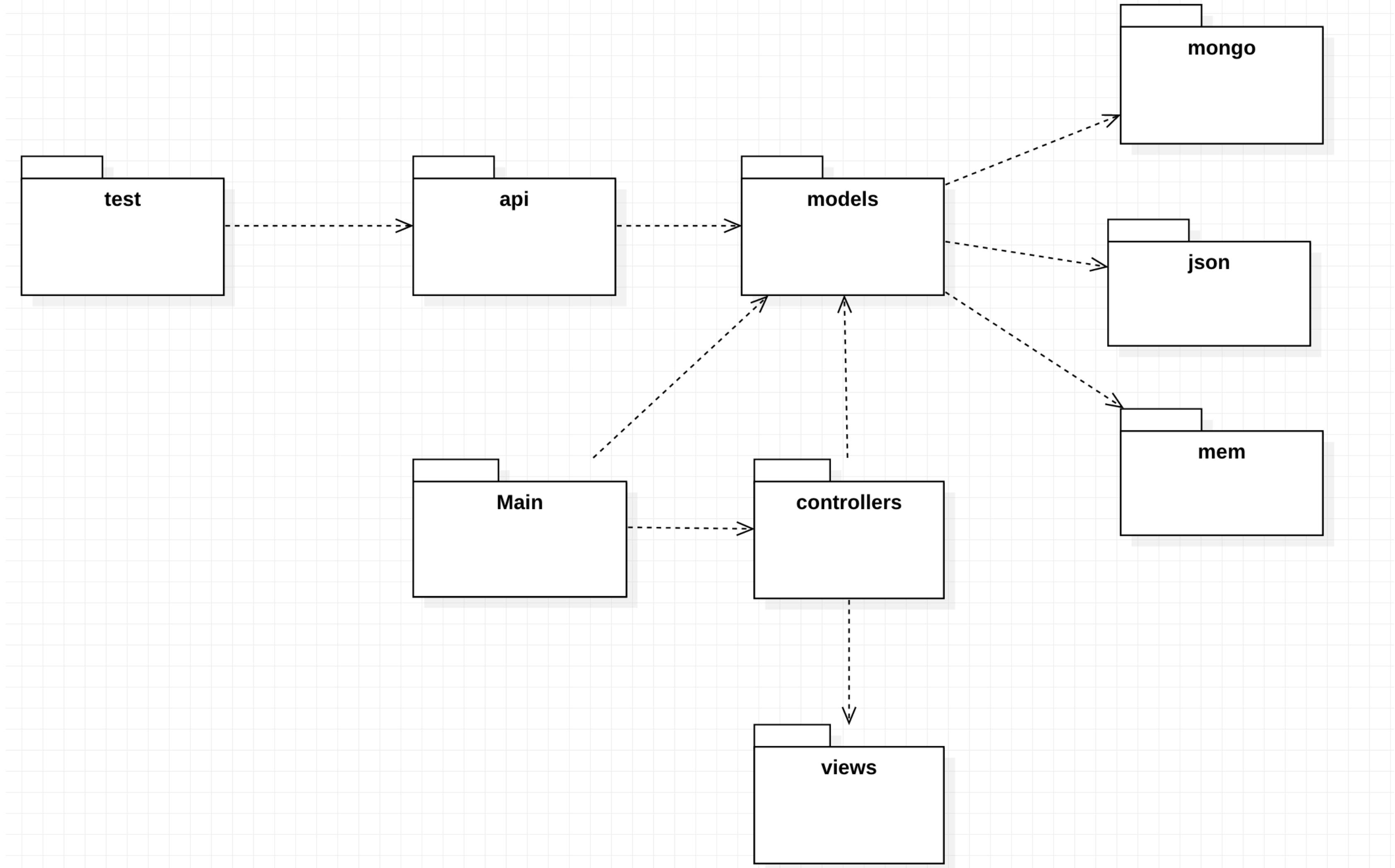


## Object Diagram Elements



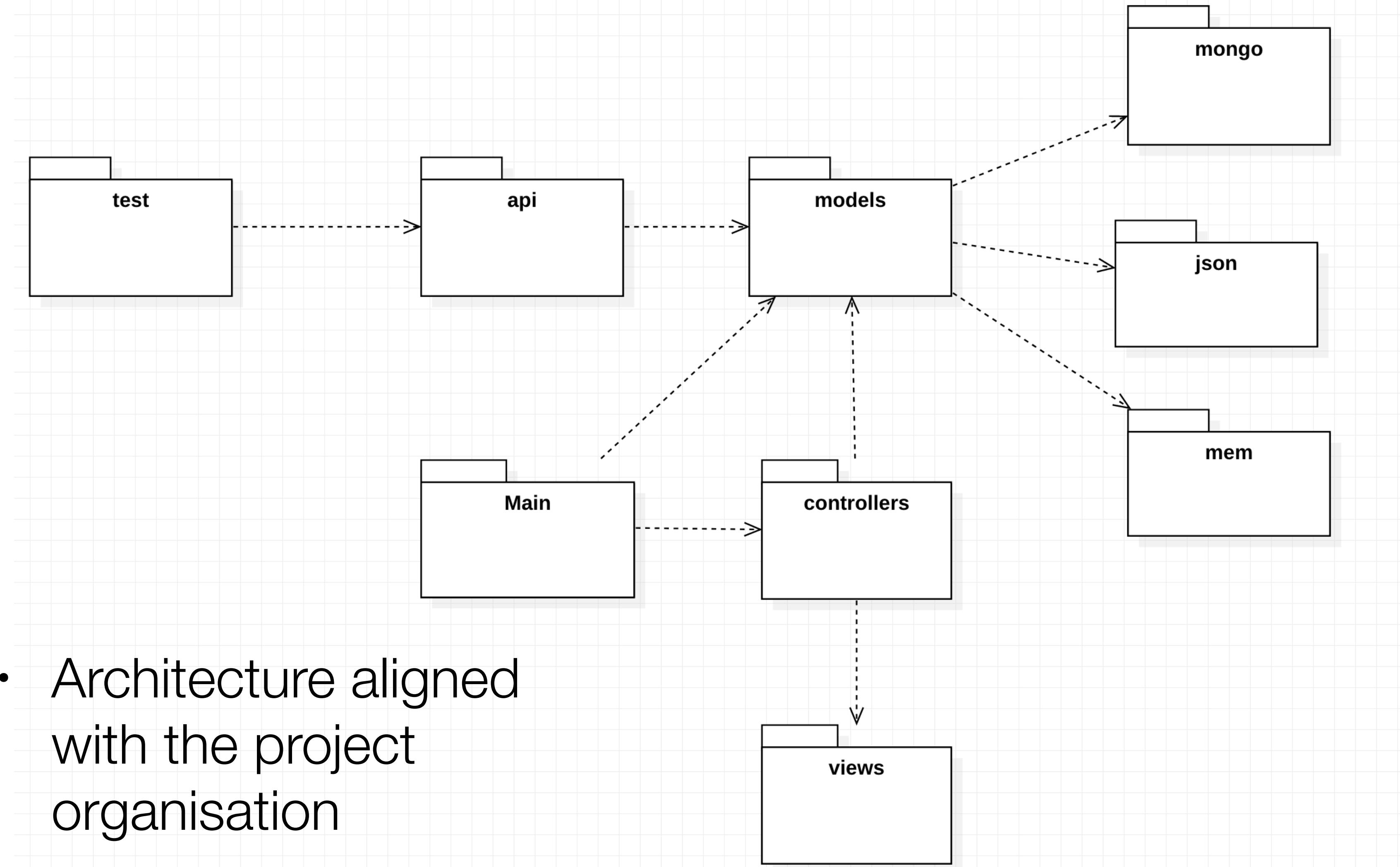
<https://loufranco.com/wp-content/uploads/2012/11/cheatsheet.pdf>

# Playtime Architecture



```
src
  > api
  > controllers
  < models
    > json
    > mem
    > mongo
      JS db.js
      JS joi-schemas.js
  > views
    JS api-routes.js
    JS server.js
    JS web-routes.js
test
  > api
  > models
    JS fixtures.js
    JSON fixtures.json
    JS test-utils.js
.env
.eslintrc.json
.gitignore
.prettierc.json
CHANGELOG.md
package-lock.json
package.json
```

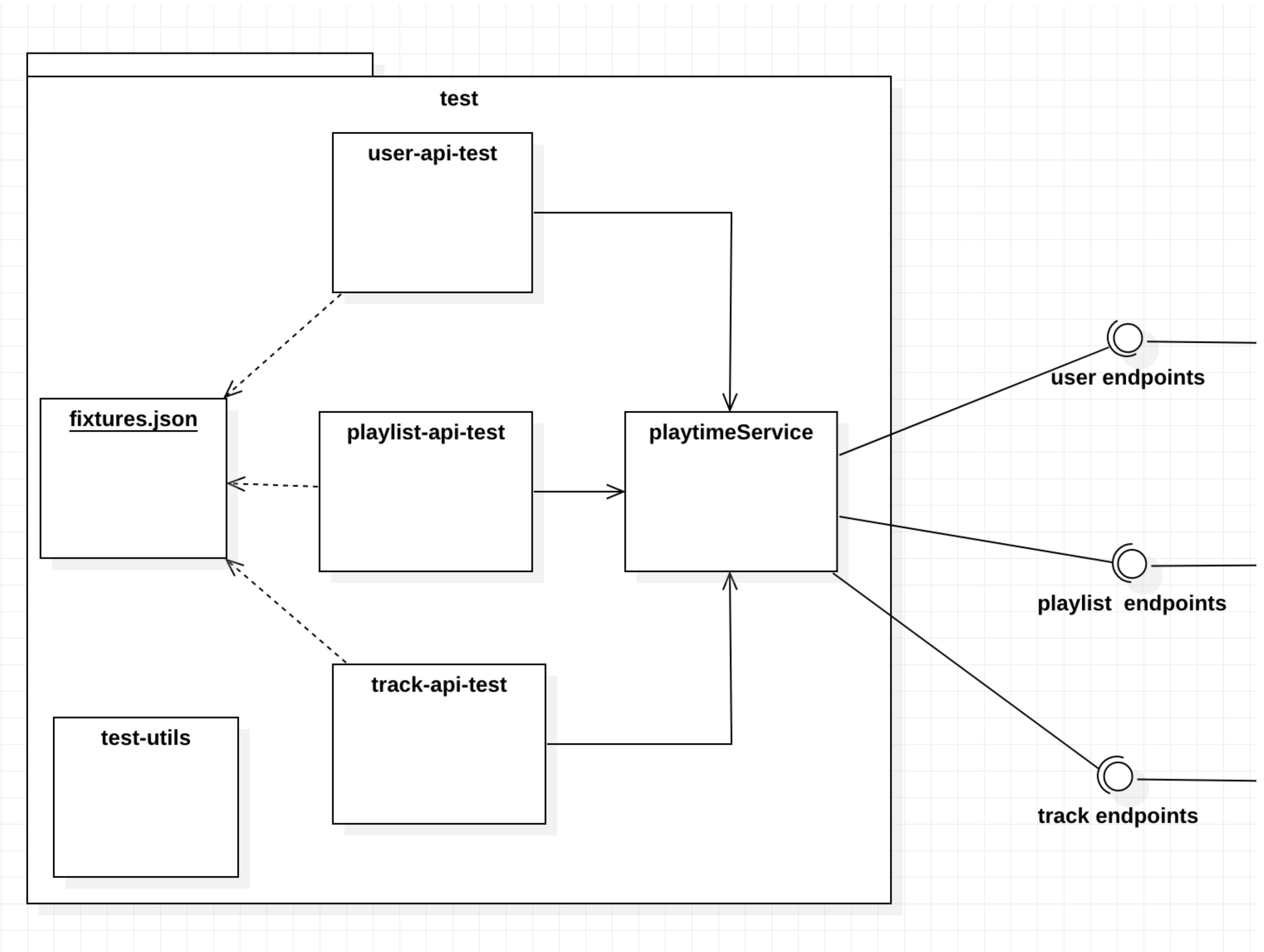
# Playtime Architecture



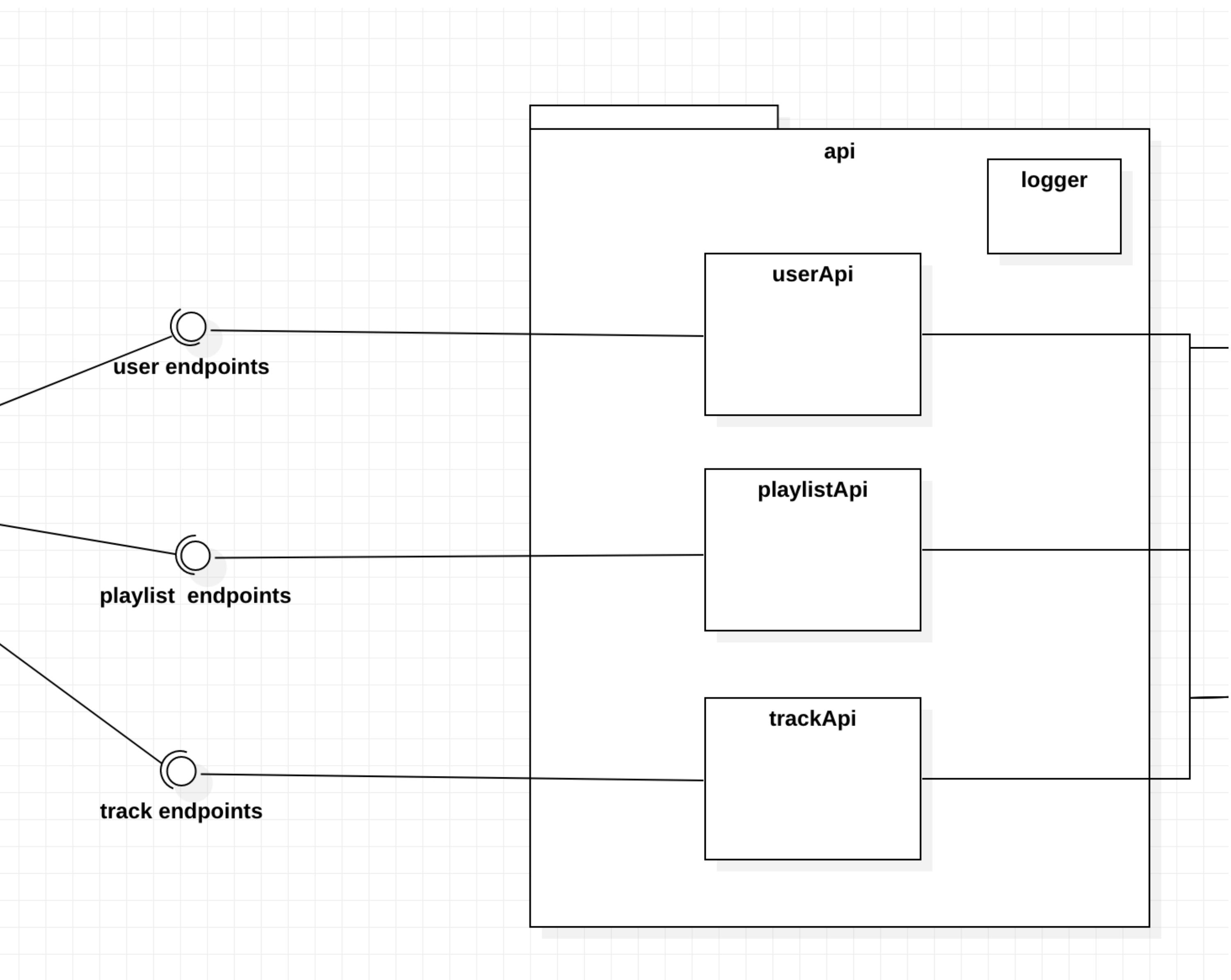
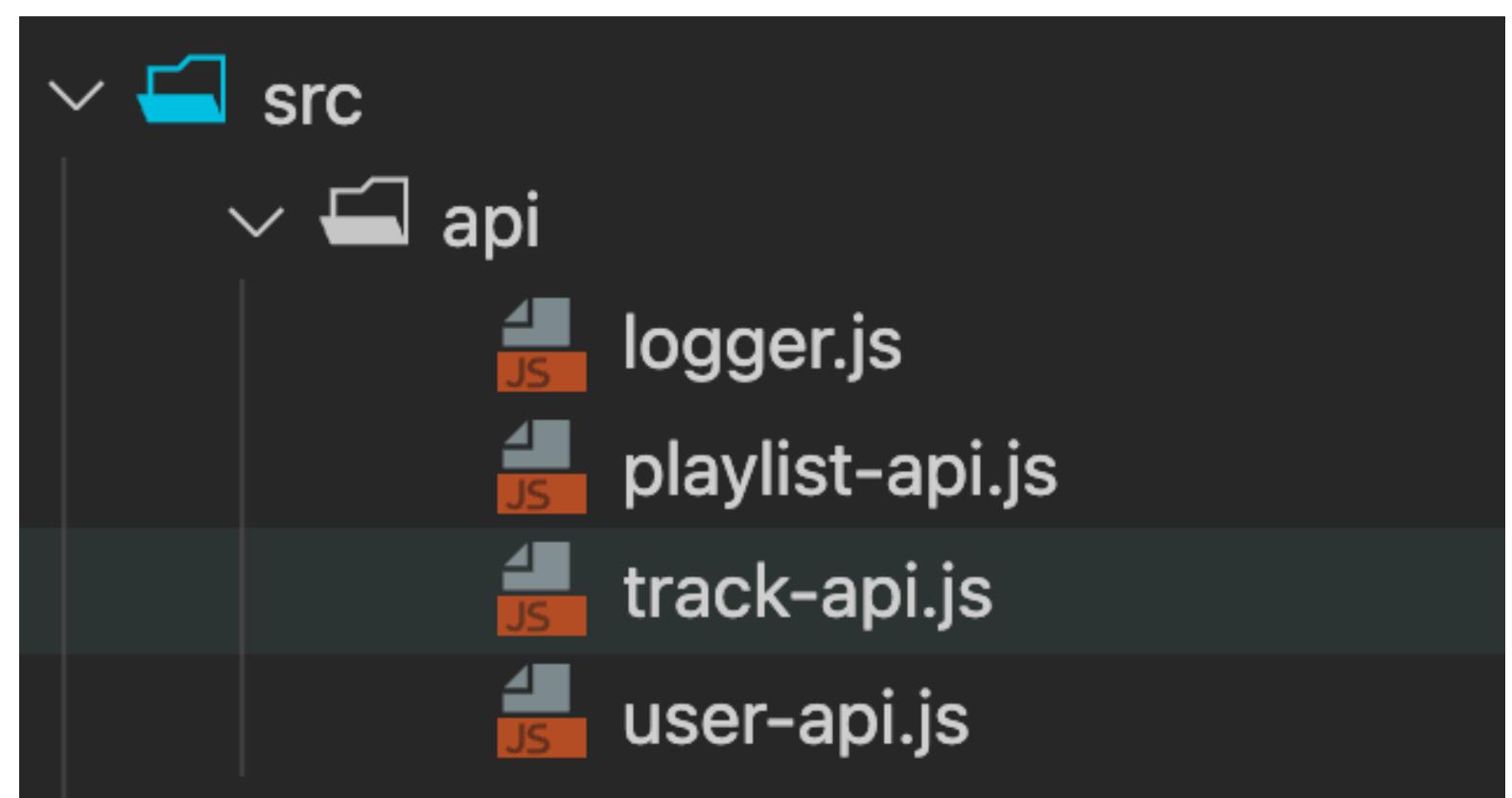
- Architecture aligned with the project organisation

# Test

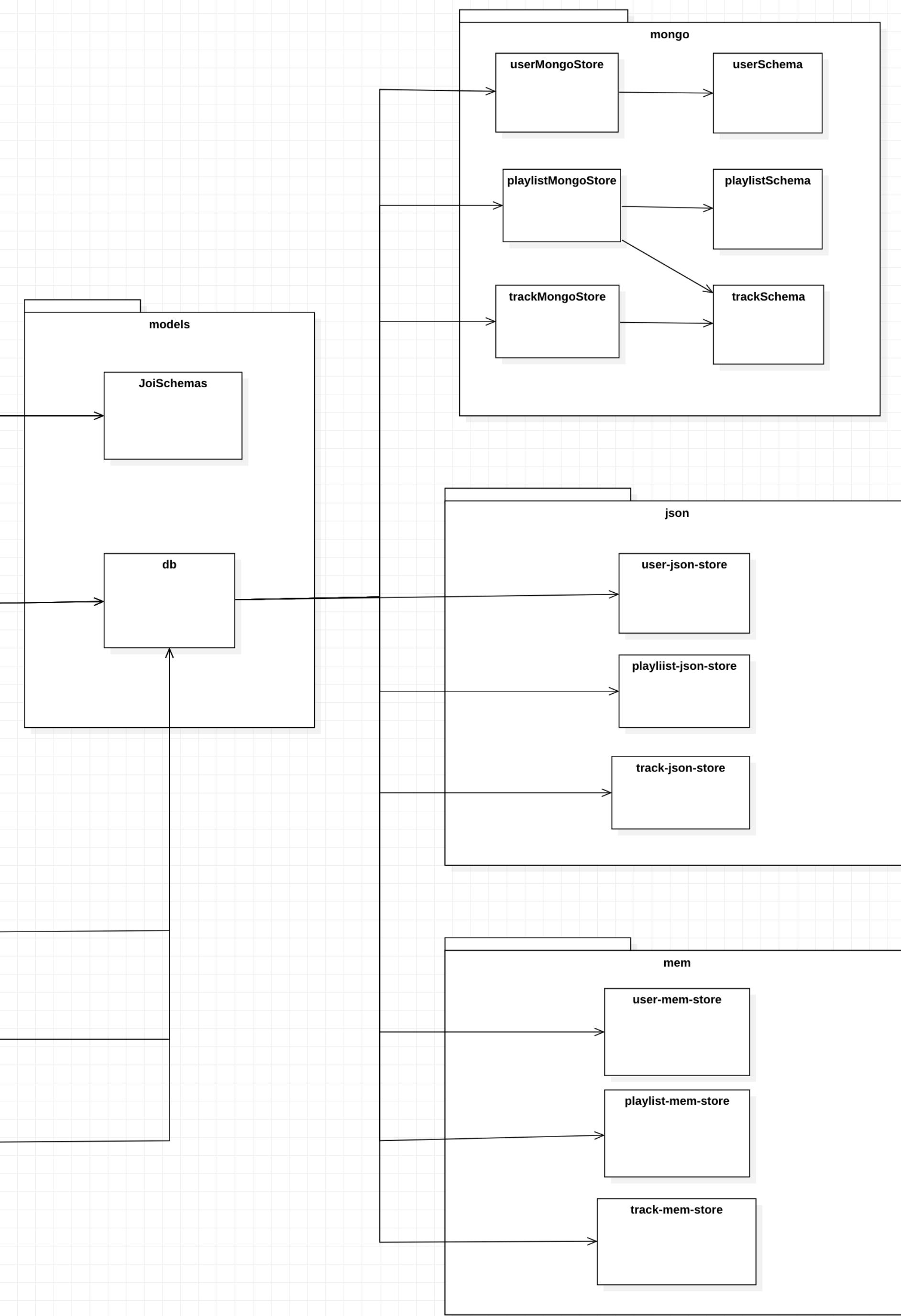
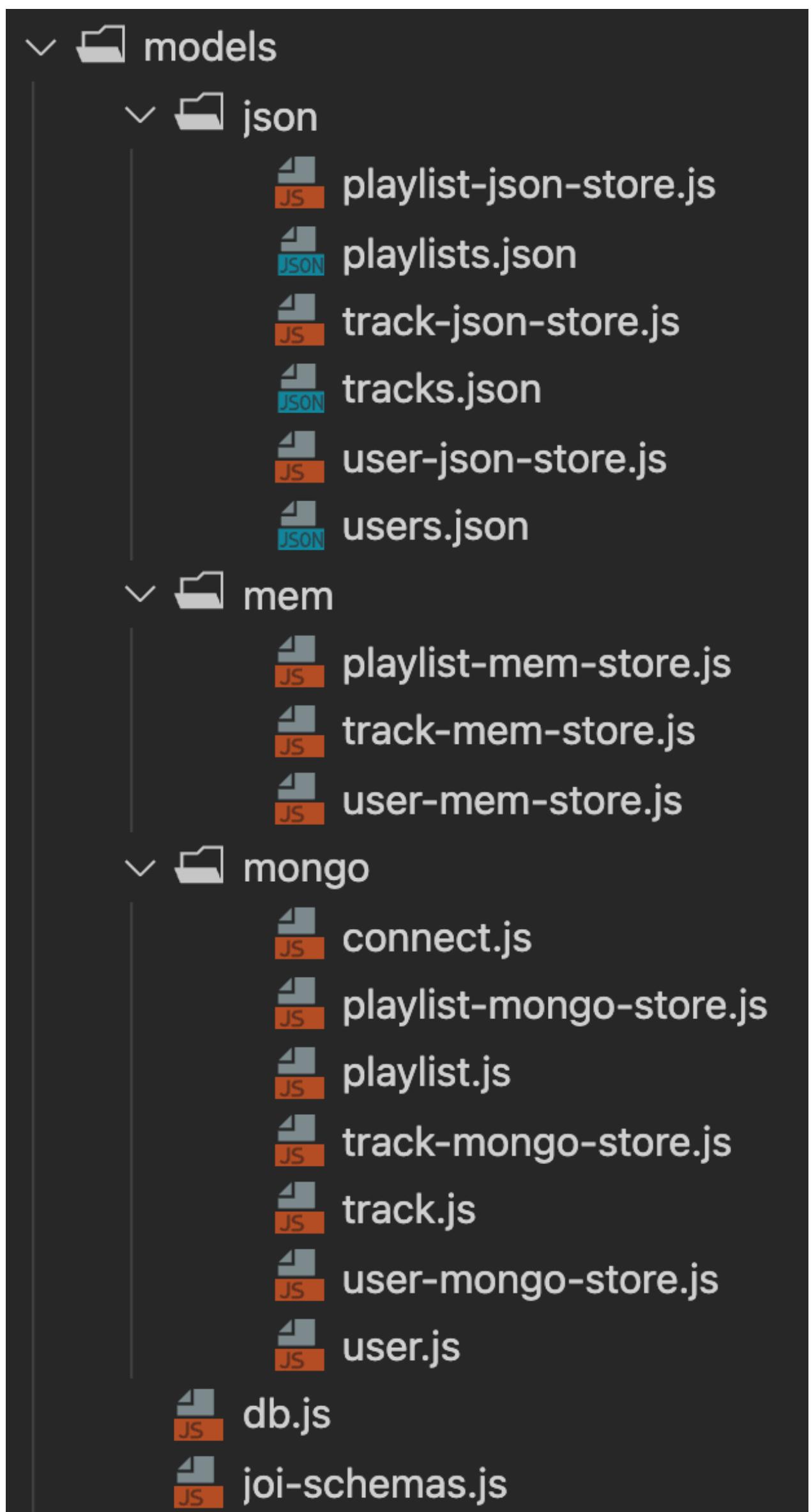
```
test
  api
    playlist-api-test.js
    playtime-service.js
    tracks-api-test.js
    users-api-test.js
  models
  fixtures.js
  test-utils.js
```



# Api



# Models



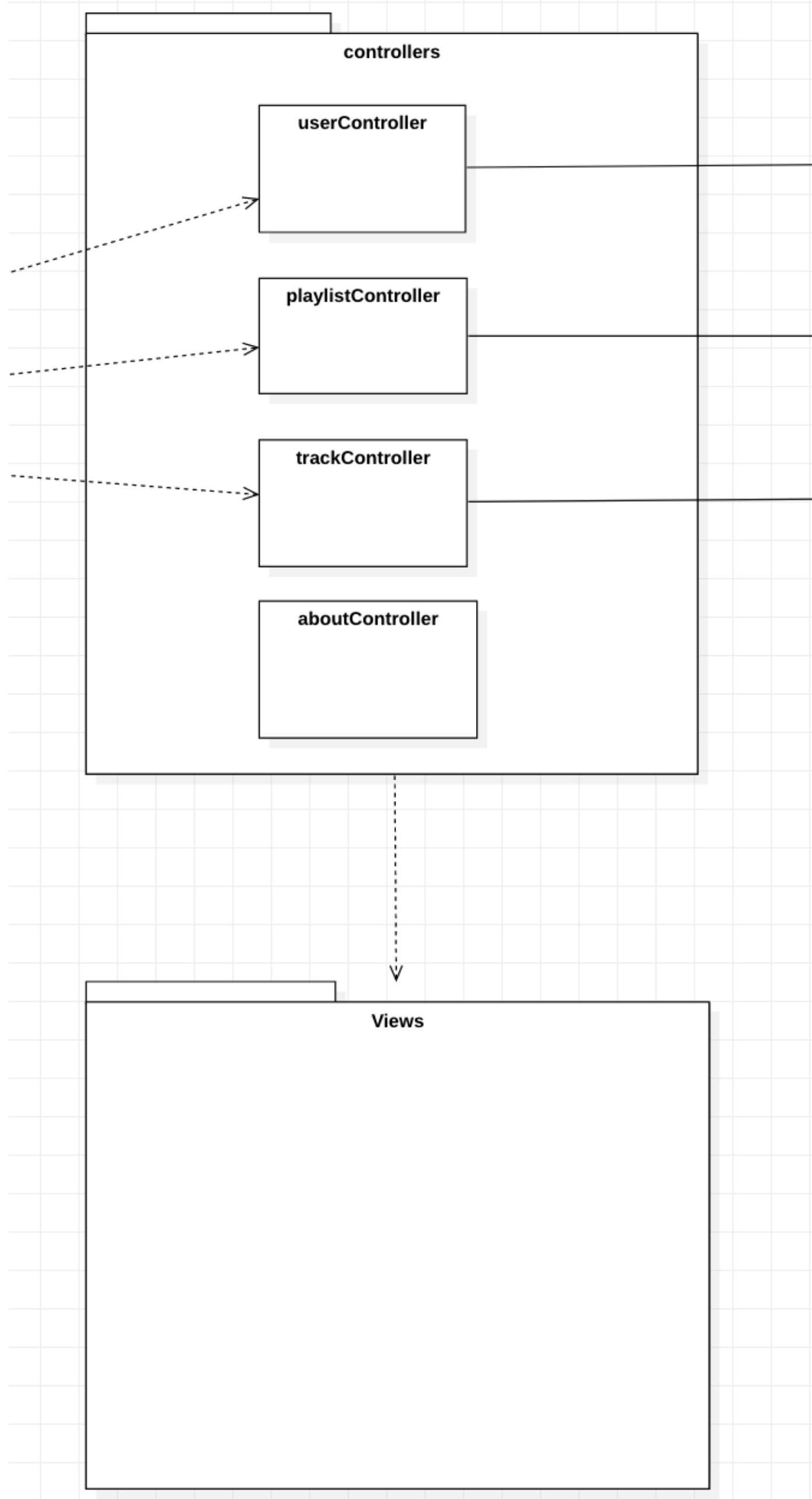
# Controllers/Views

```
✓ └── controllers
    ├── about-controller.js
    ├── accounts-controller.js
    ├── dashboard-controller.js
    ├── playlist-controller.js
    └── track-controller.js

✓ └── views
    └── layouts
        └── layout.hbs

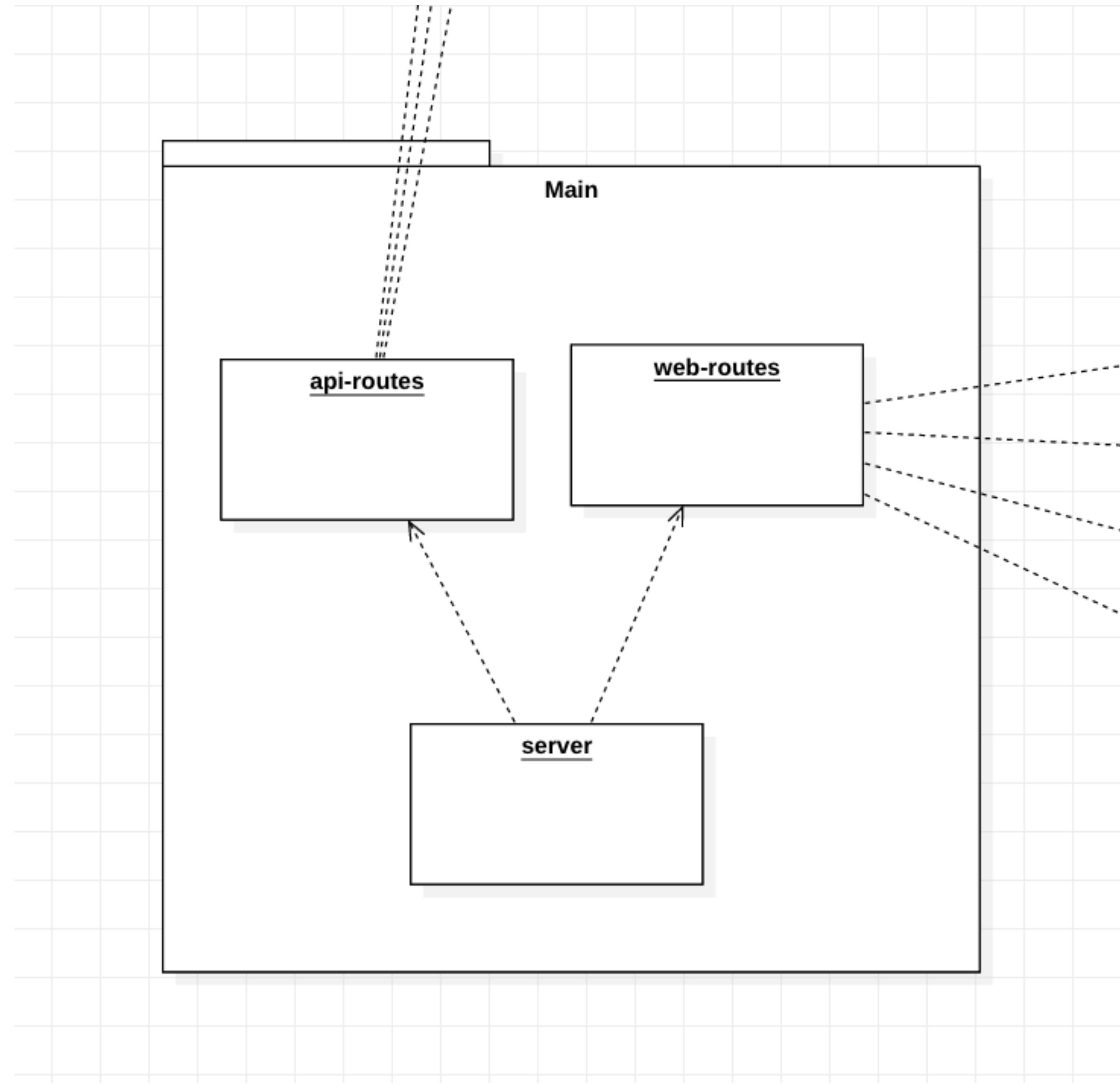
    └── partials
        ├── add-playlist.hbs
        ├── add-track.hbs
        ├── error.hbs
        ├── list-playlists.hbs
        ├── list-tracks.hbs
        ├── menu.hbs
        ├── playtime-brand.hbs
        └── welcome-menu.hbs

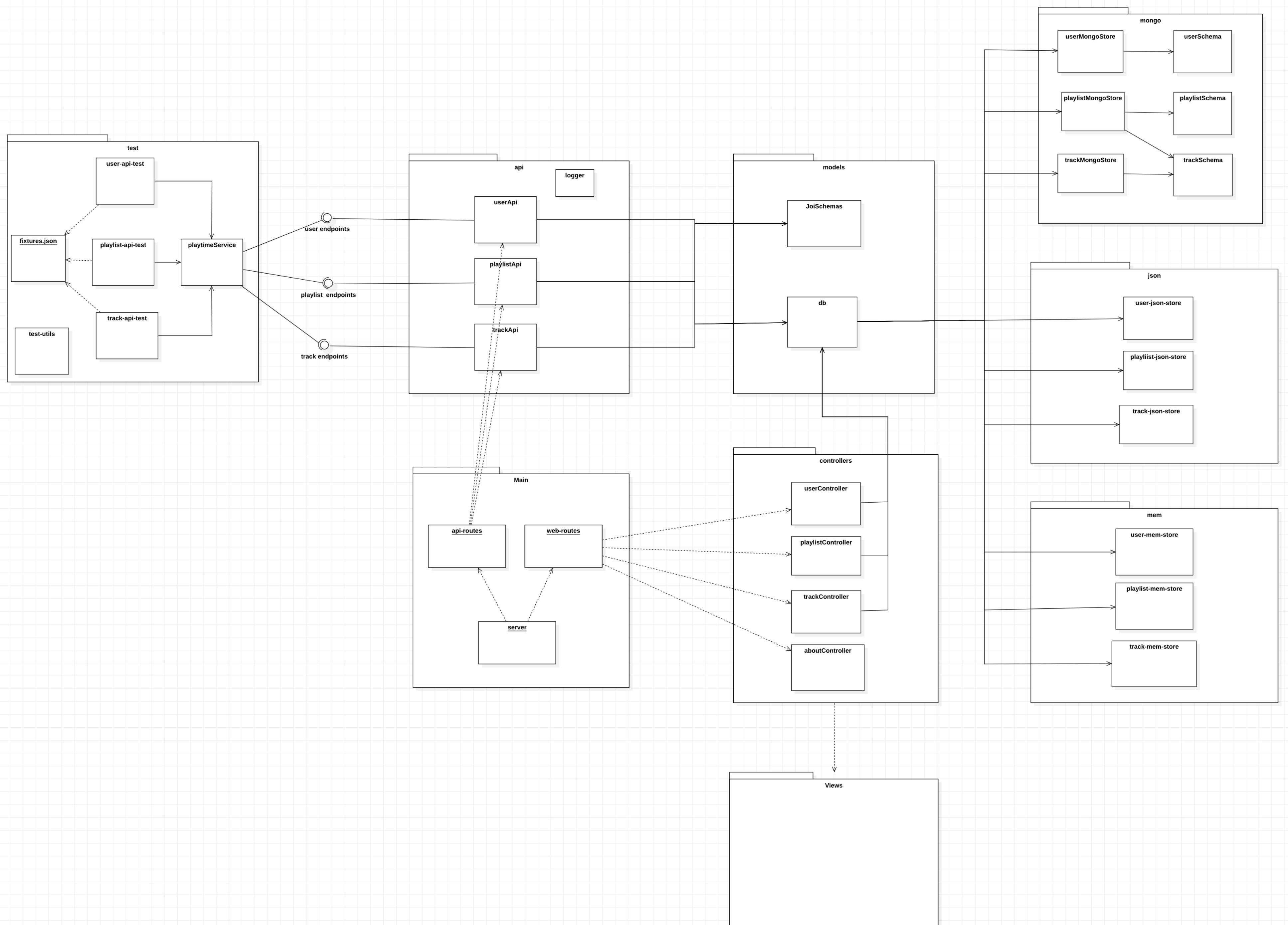
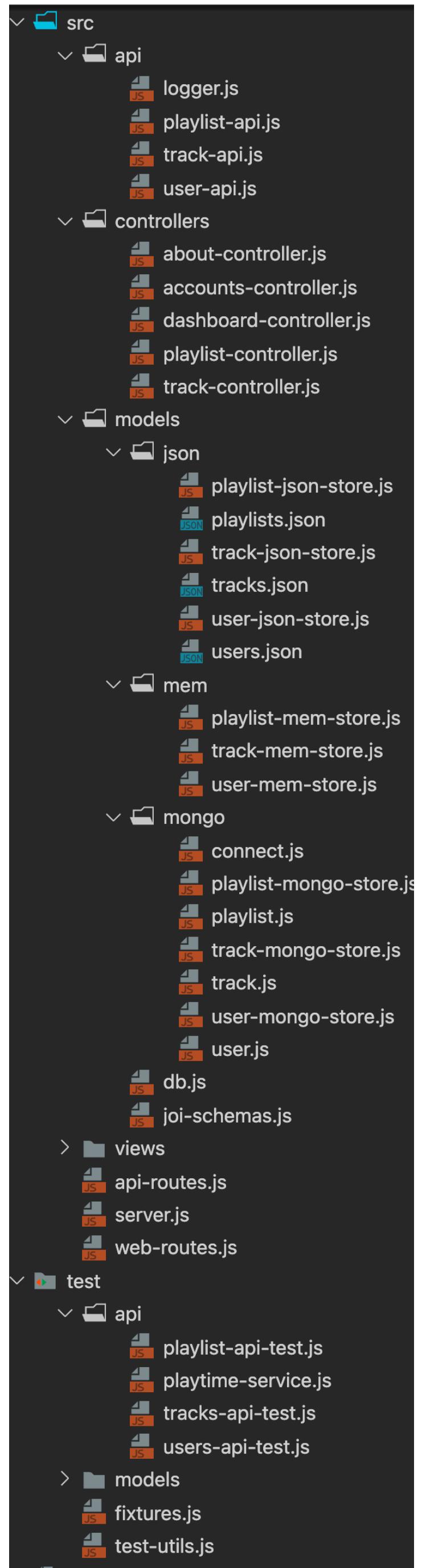
        └── about-view.hbs
        ├── dashboard-view.hbs
        ├── login-view.hbs
        ├── main.hbs
        ├── playlist-view.hbs
        ├── signup-view.hbs
        └── track-view.hbs
```



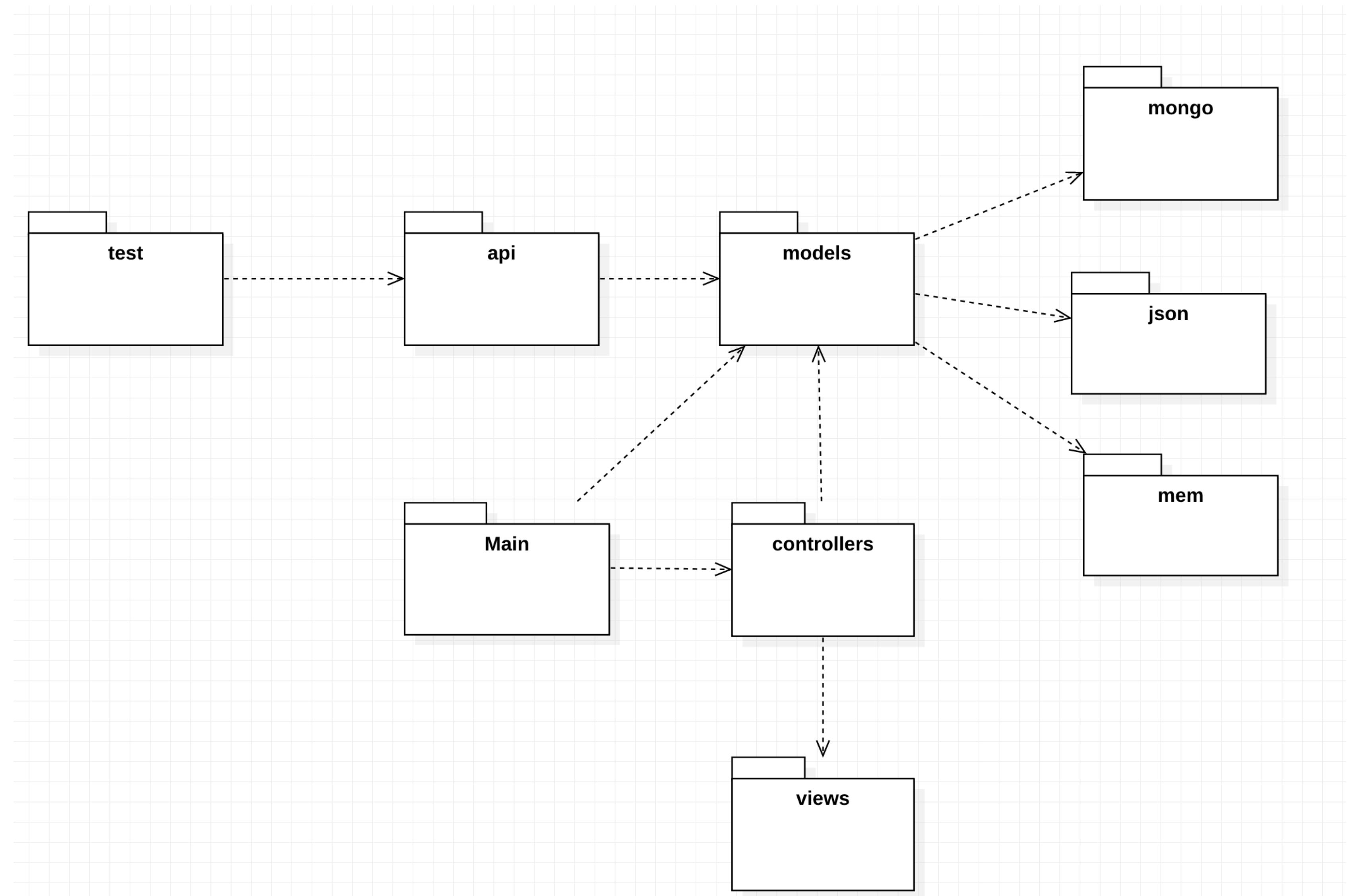
# Main

```
api-routes.js  
server.js  
web-routes.js
```



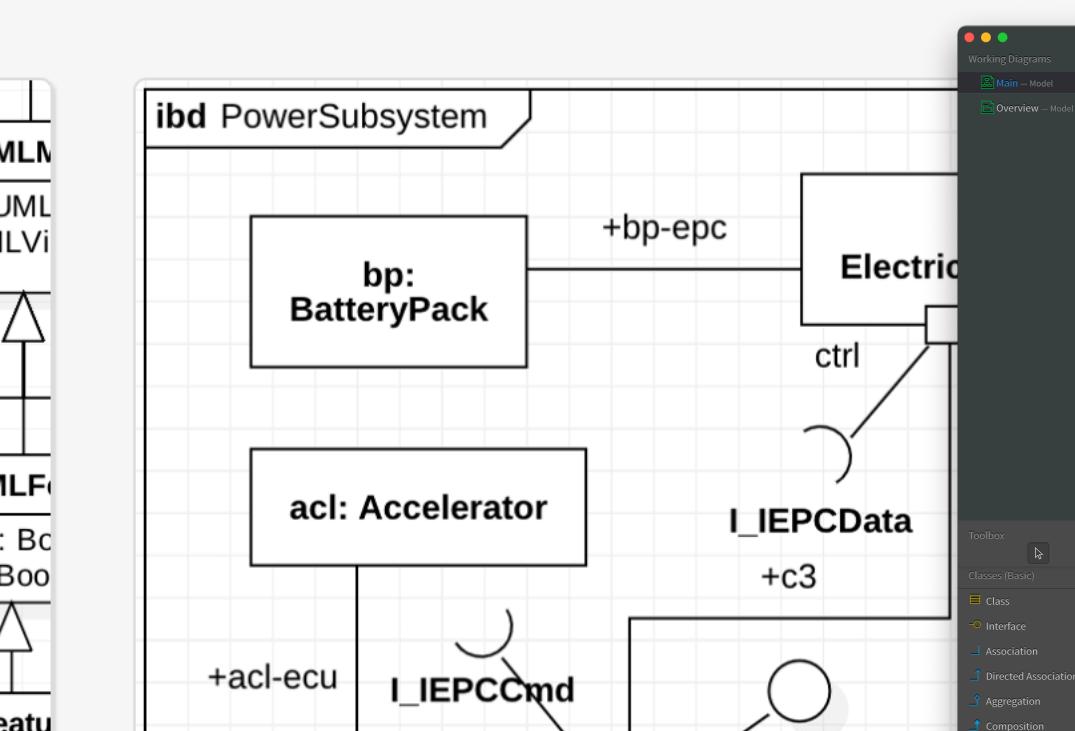
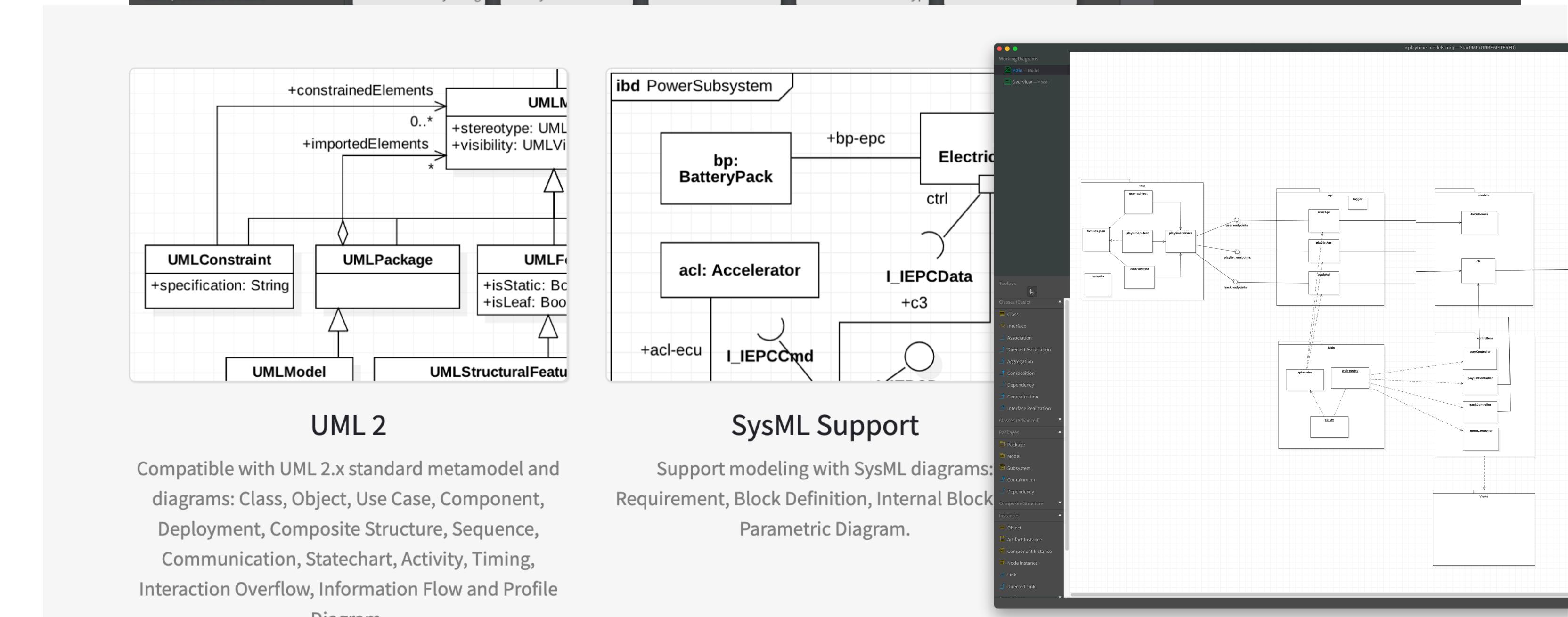
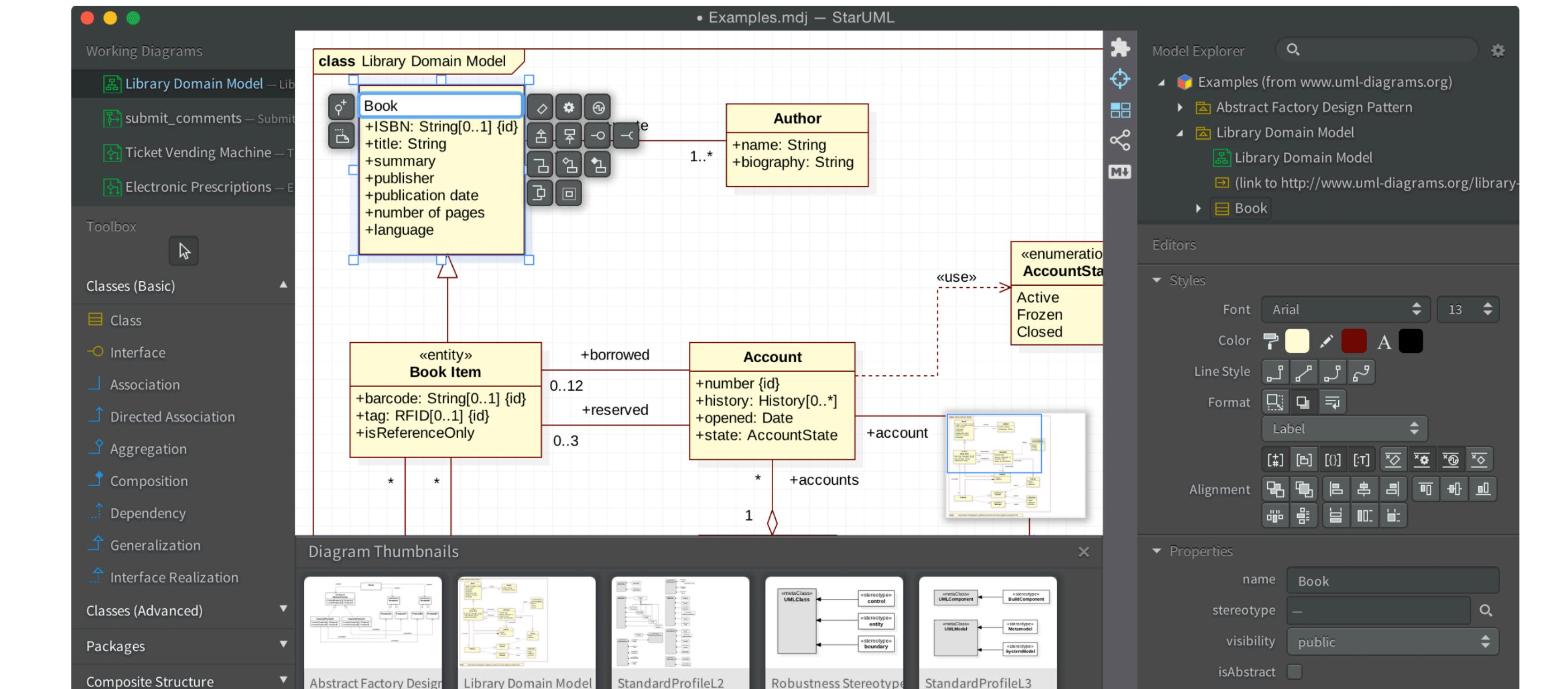


```
src
  > api
  > controllers
  < models
    > json
    > mem
    > mongo
      JS db.js
      JS joi-schemas.js
  > views
    JS api-routes.js
    JS server.js
    JS web-routes.js
  test
    > api
    > models
      JS fixtures.js
      JSON fixtures.json
      JS test-utils.js
  .env
  .eslintrc.json
  .gitignore
  .prettierrc.json
  CHANGELOG.md
  package-lock.json
  package.json
```



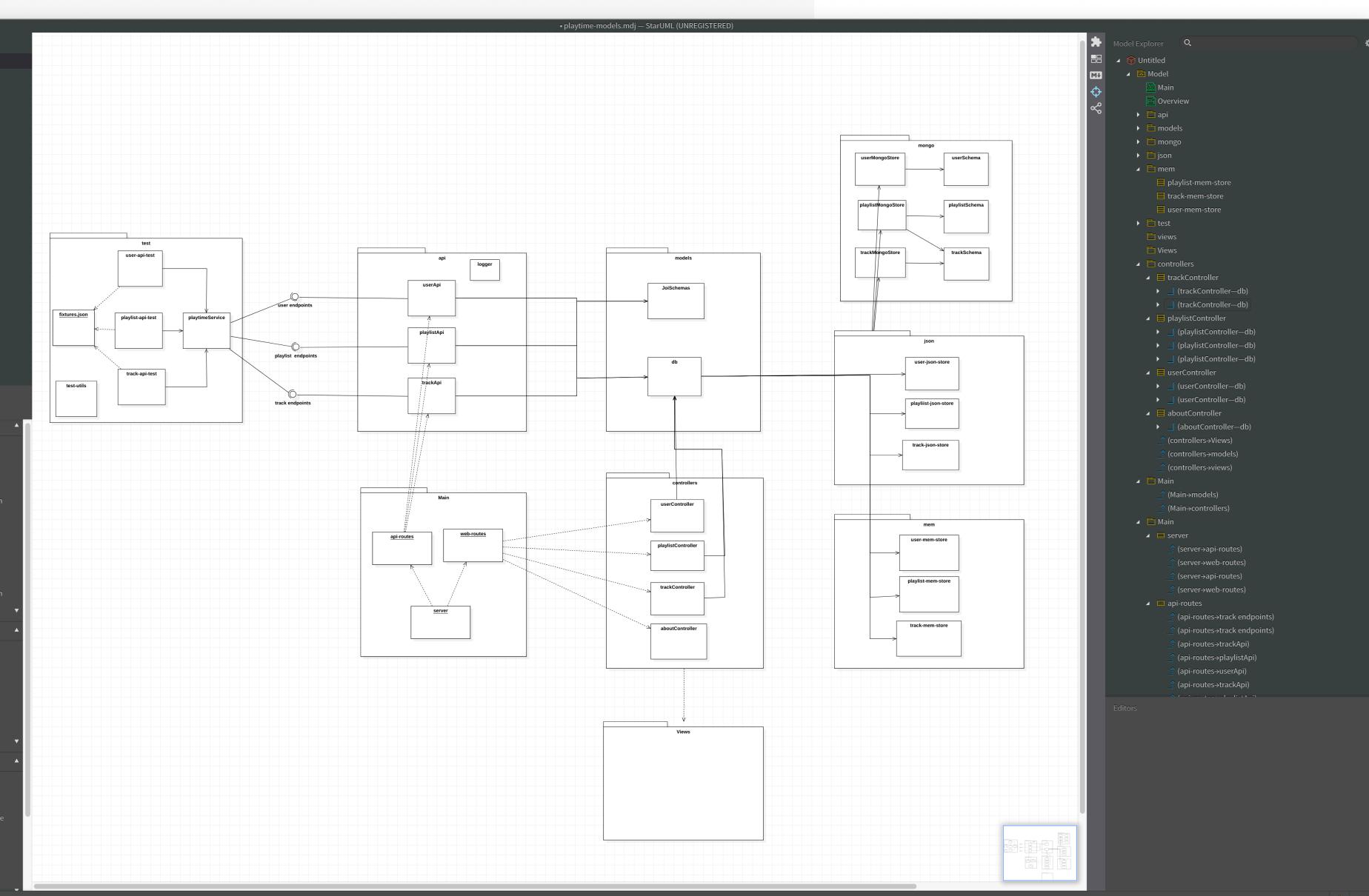


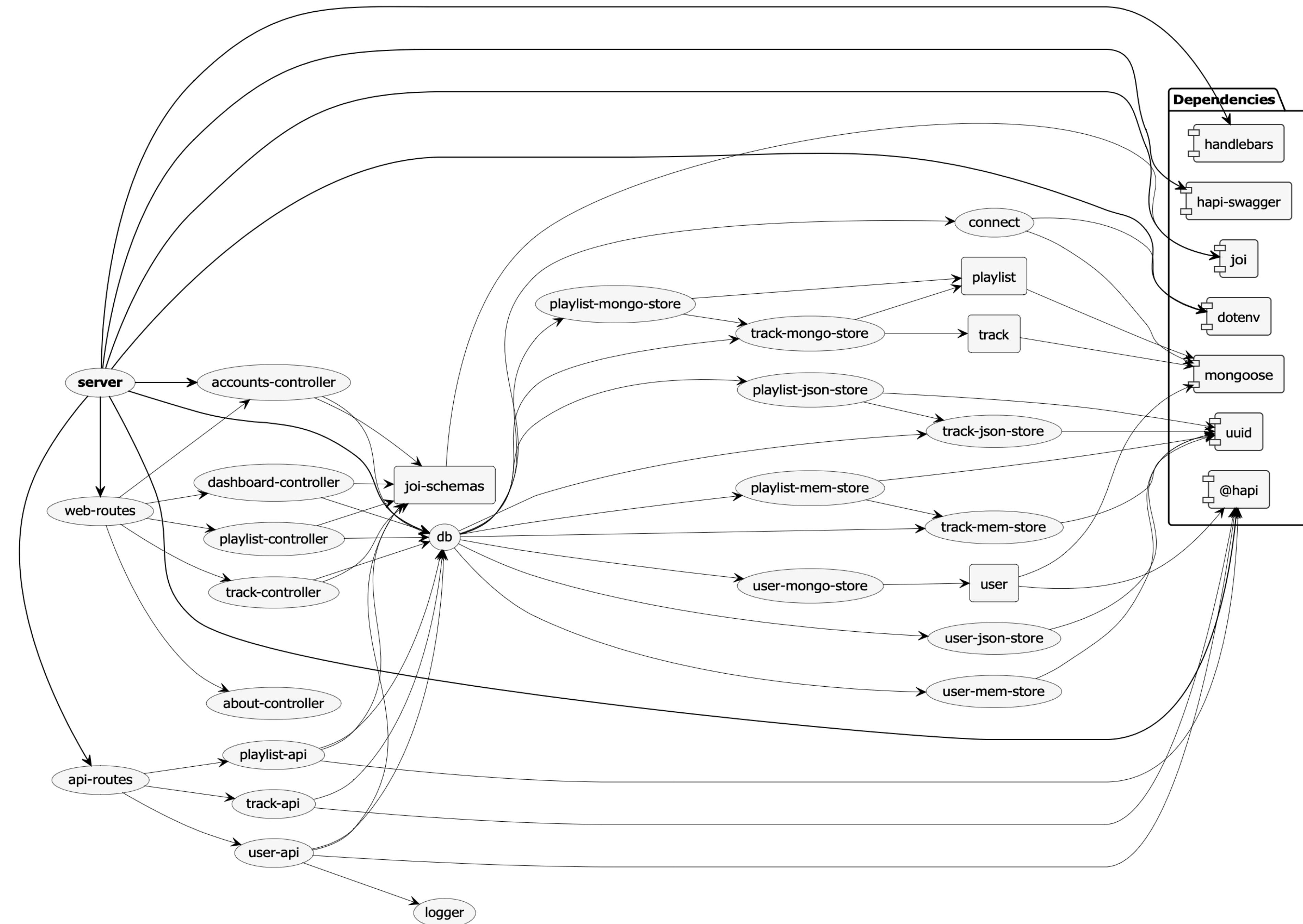
# StarUML



# SysML Support

Support modeling with SysML diagrams:  
Requirement, Block Definition, Internal Block  
Parametric Diagram.







arkitektur

v1.6.4 downloads 2k/week dependents 34 build unknown coverage test report not found technical debt 10%

# Visualises JavaScript, TypeScript and Flow codebases as meaningful and committable architecture diagrams

- Associates source files with configured architectural components
- Renders grouped components and dependency graph including Node.js modules
- Supports JavaScript, TypeScript and Flow source code and Vue/Nuxt
- Exports codebase architecture visualisation as SVG, PNG or Plant UML diagram
- Integrates into development flow, so your CI, VCS, README and PRs are happy

## Usage

```
# Run arkit straight away
npx arkit
```

```
# Or add it to your project as a dev dependency
npm install arkit --save-dev
yarn add arkit --dev
```

```
# Run arkit against your source folder and save result as SVG
npx arkit src/ -o arkit.svg
```

```
# You can also specify source files to start from and output format
npx arkit -f src/main.js -o puml
```

```
# And get some more with debugging and file exclusions
LEVEL=info npx arkit -e "node_modules,test,dist,coverage" -o puml
```

If your project is huge and first diagrams look messy, it's better to generate them per feature, architectural layer, etc.

Once you satisfied with results, add arkit command to your build script, so it will keep your architecture diagrams up-to-date.

# structure101

PUT YOUR  
SOFTWARE STRUCTURE  
TO WORK

DOWNLOAD NOW



## Visualize

Visualizing the structure that emerges from source code to instantly boost architectural accuracy and developer productivity.



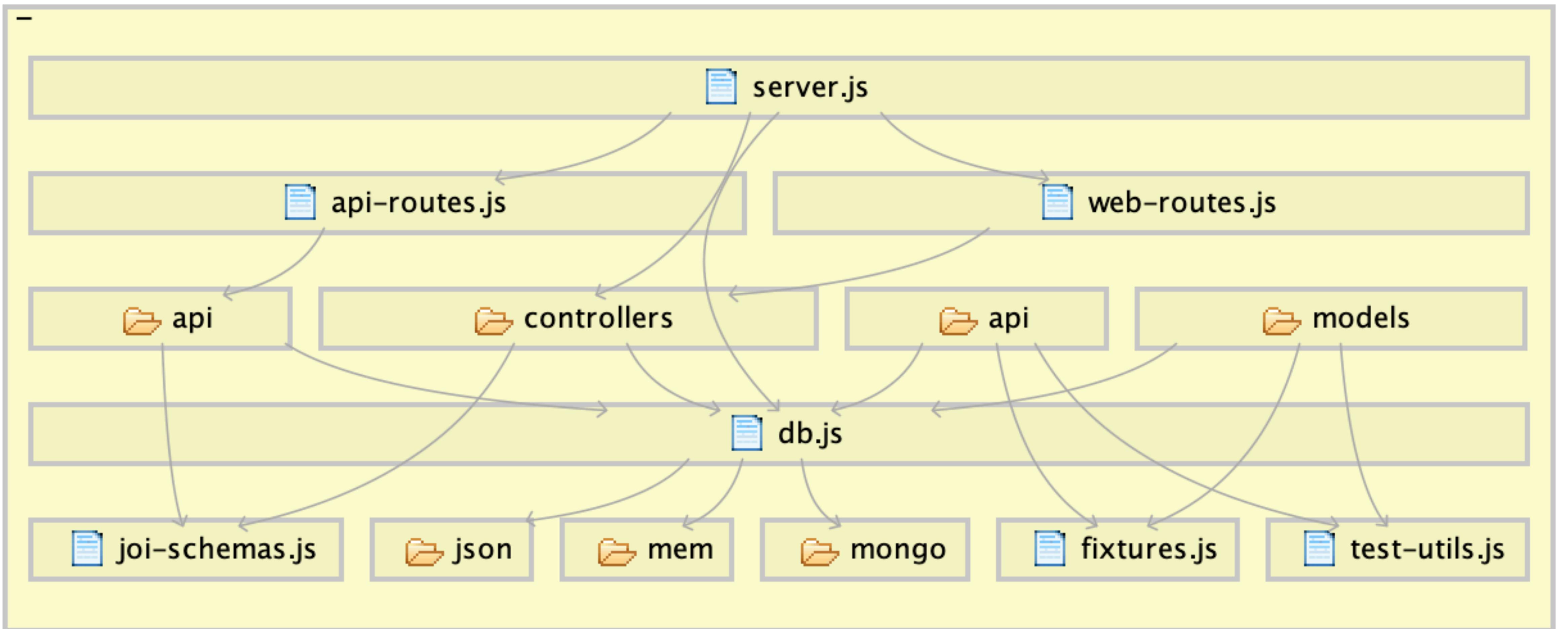
## Specify

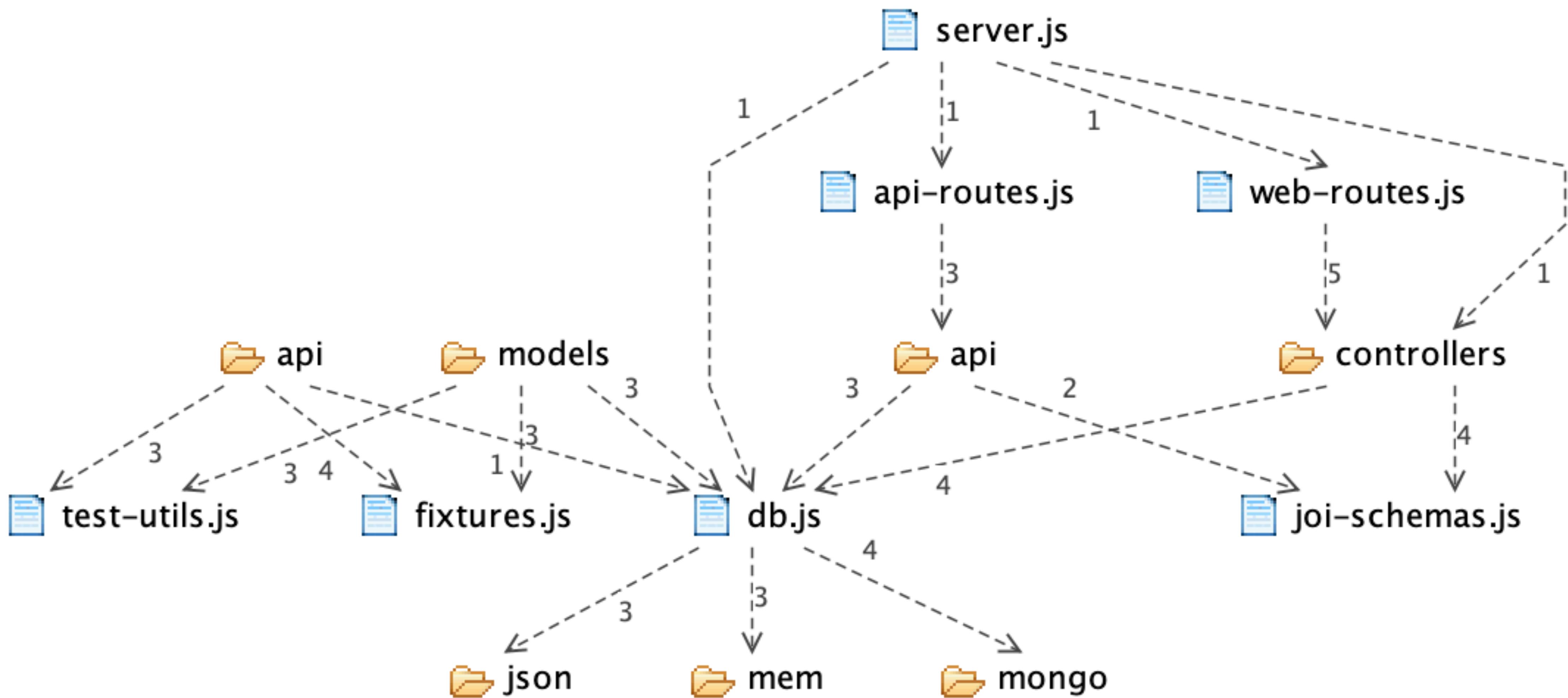
Add API and dependency specs to the emergent model to guide the team as they edit code, and to catch violations at build time.

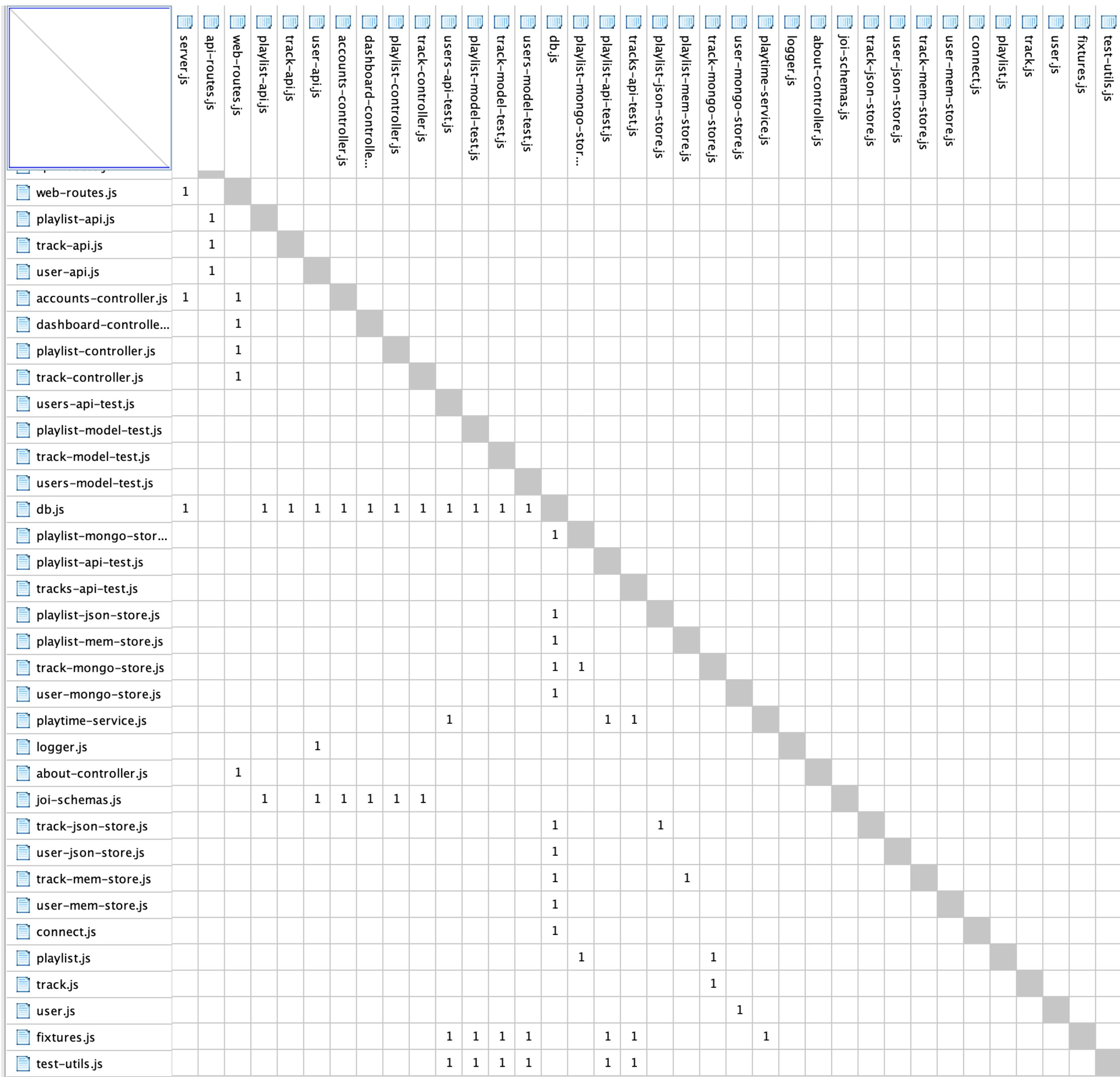


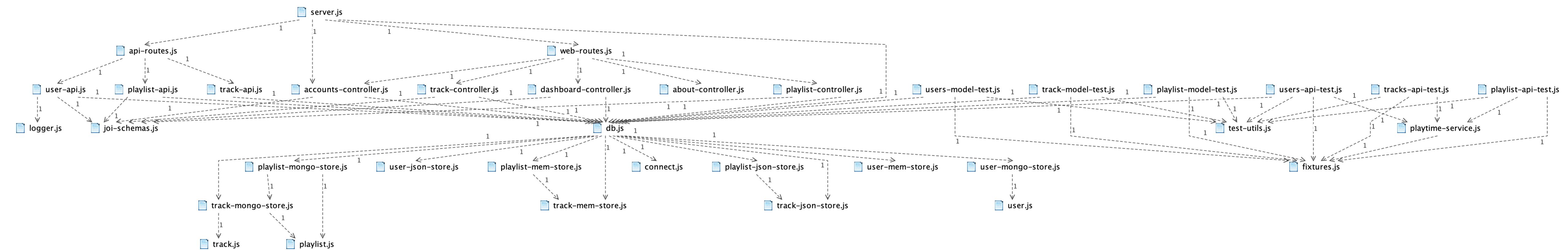
## Simplify

Manipulate the visual model to disentangle monoliths, making the visualized, specified codebase more modular and even easier to adapt and develop.

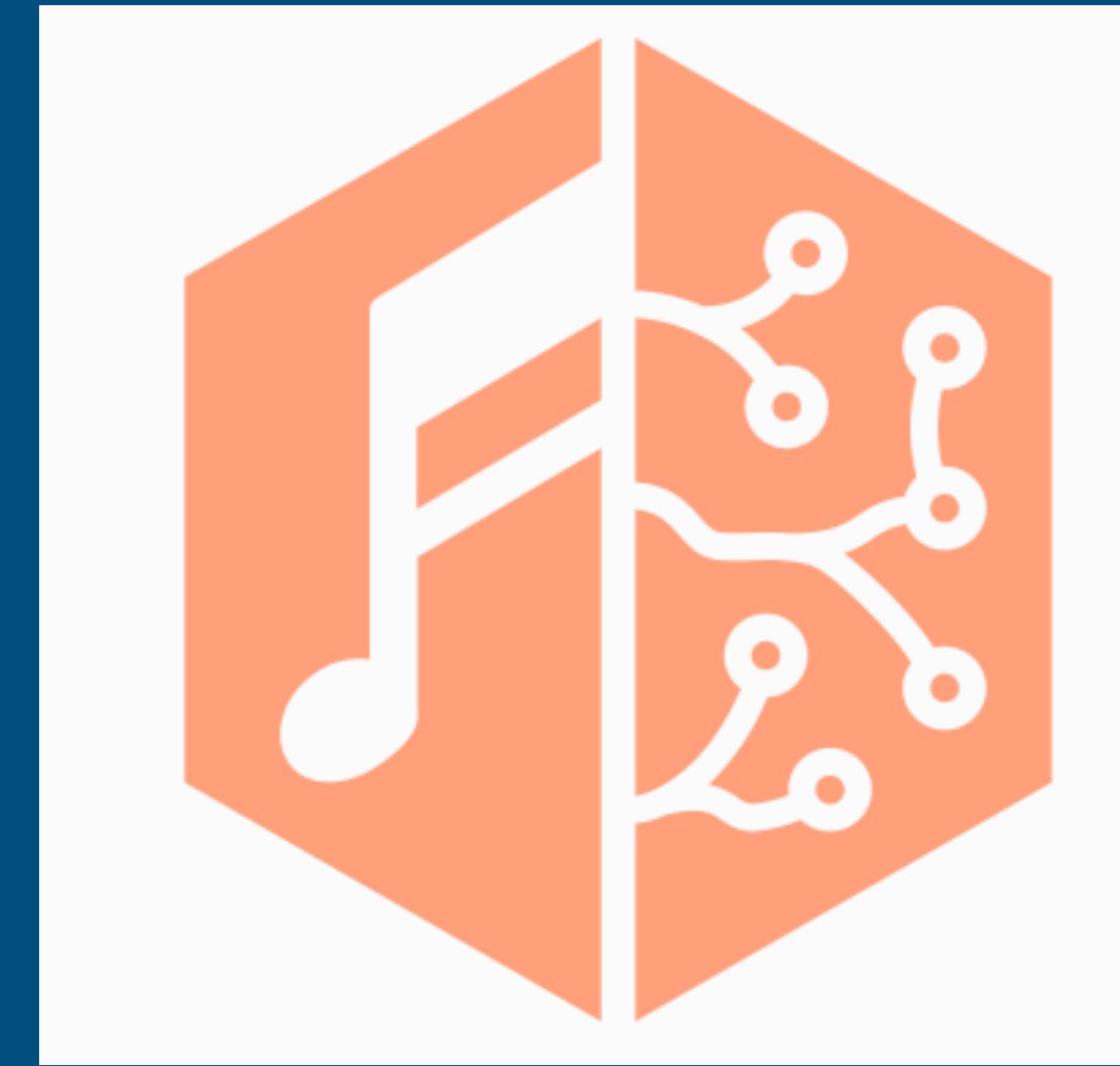








# Playtime Architecture



Full Stack Web Development