

Empowering remote education with the Tutors Open Source Project

tutors.dev



TutorStack

Coherent toolset for teaching online

Layer 1	Learning/Instruction materials	 Tutors
Layer 2	Communications & Community	 slack  zoom
Layer 3	Assessment & Feedback	 moodle  socrative
Layer 4	Media Management & Streaming	 YouTube  zoom  OBS  Screenbrush



An Open Learning Web Toolkit

A collection of open source components & services supporting the creation of transformative learning experiences using open web standards.

Demo

Live

Source

Documentation

Gallery

Simulator

Reference Course
A reference course containing all supported learning objects

Search | Layout | More

Simple: Units with presentations, labs + resources

Sidebar: Presentations, links + labs in side bar

Videos: Video + Presentations and labs with videos

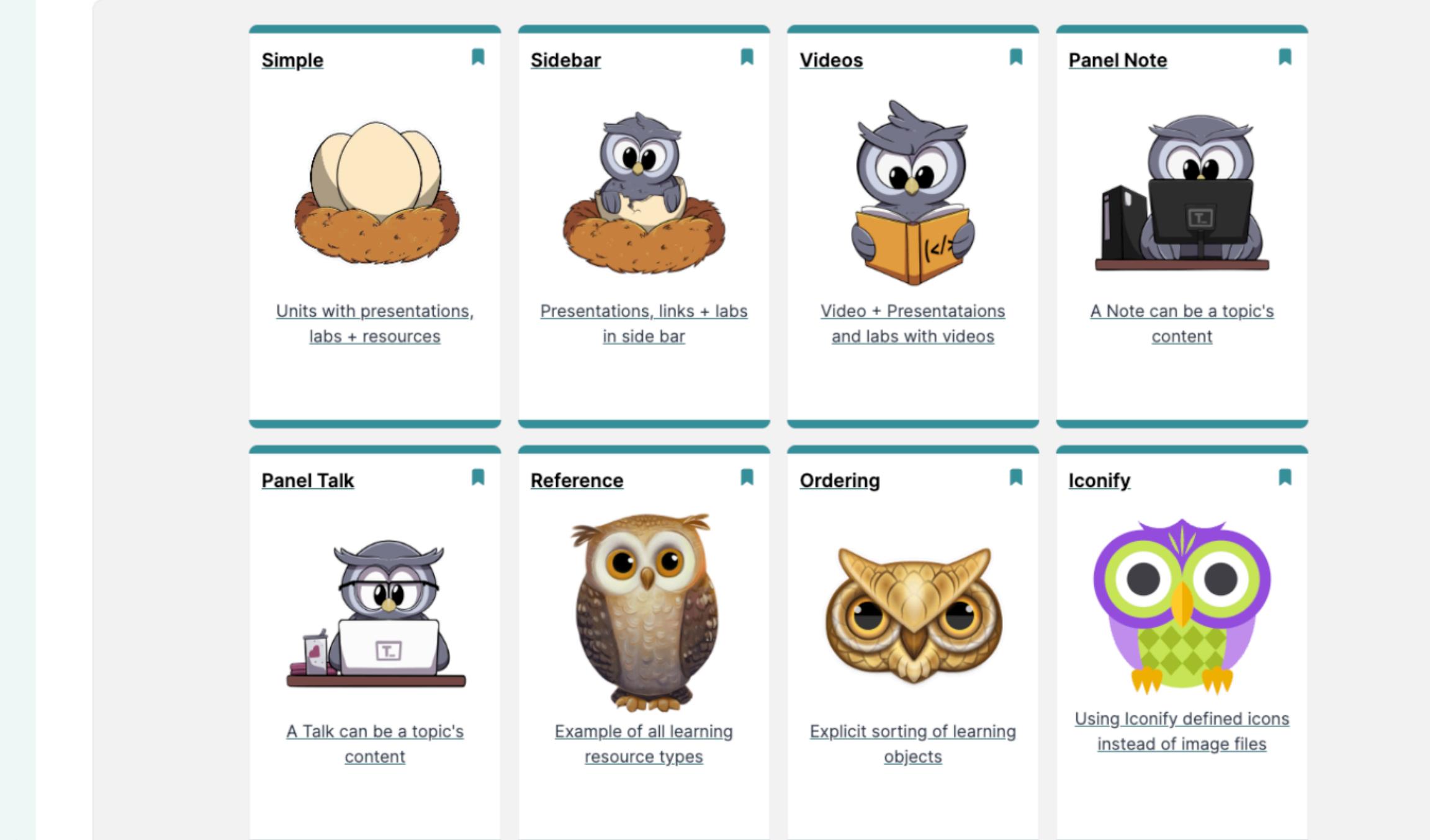
Panel Note: A Note can be a topic's content

Panel Talk: A Talk can be a topic's content

Reference: Example of all learning resource types

Ordering: Explicit sorting of learning objects

Iconify: Using Iconify defined icons instead of image files

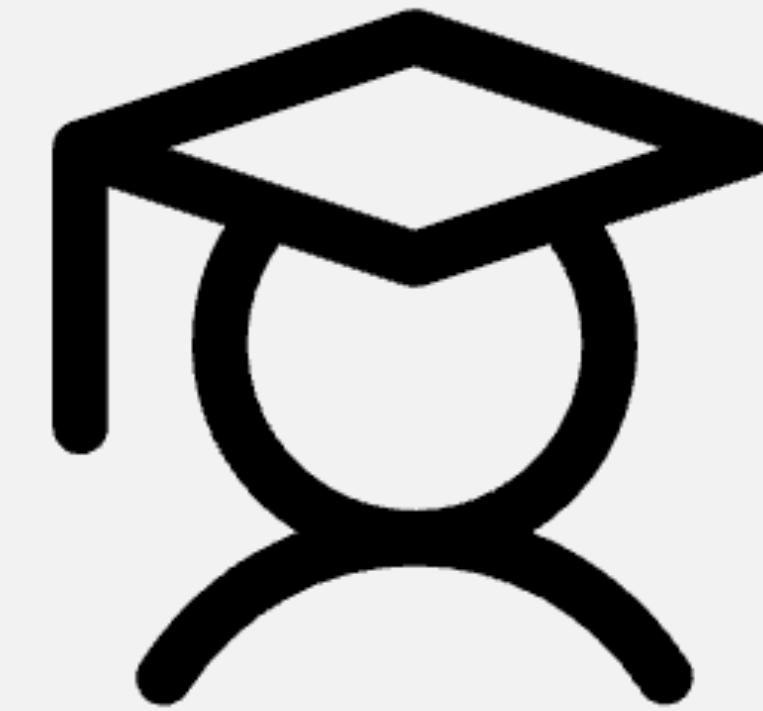


Tutors

An open web learning toolkit which supports 3 complimentary perspectives.

- Learner Experience
- Educator Experience
- Developer Experience

Learner Experience



The Learner Experience

The ***Learner Experience*** prioritises web interactions that are **engaging, contextual, linkable, searchable, accessible** and **responsive**. In addition the experience should foster a sense of **community** and **connection** among fellow learners.

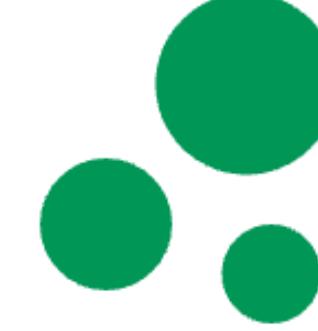
A programme homepage

Higher Diploma in Computer Science 2022
Department of Computing & Mathematics, WIT

Layout

Semester One

Workshop One



induction · structure · schedules · handbook

Life etc...



time · space · balance · experiences · tips · techniques

Web Development I



html · css · layout · web apps · web frameworks · deployment

Programming



algorithms · data structures · processing · java · classes · libraries

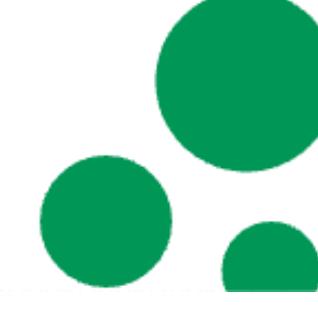
Semester 1 Schedule



timetables · semester calendars · assessment schedules

Semester 2: June-December 2020

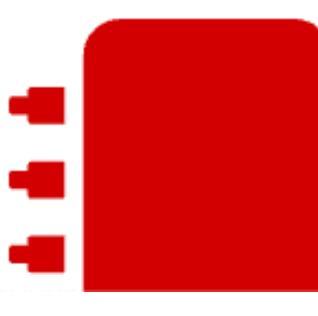
Workshop Two



Web Development II



Computer Systems & Networks



Database



Semester 2 Schedule

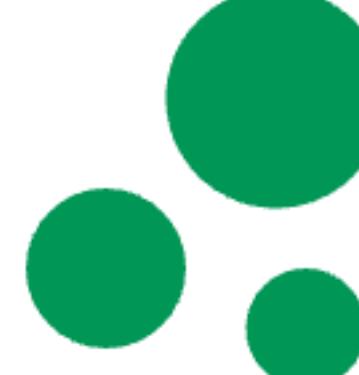


Links

Semester One

HDIP
23

FAQ 
Questions and Answers for new students on the programme

Workshop One 
induction · structure · schedules · handbook

Life etc... 
time · space · balance · experiences · tips · techniques

Web Development I 
html · css · layout · web apps · web frameworks · deployment

Programming 
algorithms · data structures · processing · java · classes · libraries

Semester 1 Schedule 
timetables · semester calendars · assessment schedules



Links to individual module
Home pages

A course homepage

The screenshot shows a course homepage for "Full Stack Web Development" by Eamonn de Leistar. The page has a light blue header with the course title and instructor's name. Below the header is a navigation bar with icons for search, layout, and user profile. The main content area displays 11 course modules arranged in two rows of five. Each module card includes an icon, the module number, a title, and a brief description.

Module	Title	Description
01: Node	node.js	The fundamentals of node. A first node application.
02: Hapi.js	hapi	The Hapi application Framework
03: Joi	joi	Validation with Joi
04: TDD	TDD cycle	Test Driven Development
05: Models	MongoDB icon	Incorporate Mongo Models
06: APIs	API endpoint icon	Expose an API from an application
07: REST	RESTful API icon	Resful API implementations
08: OpenAPI	OpenAPI icon	Documentating OpenAPI/Swagger APIs
09: JWT	JWT icon	Authenticating APIs with JSON Web Tokens
10: Seeding	Seedling icon	Seeding the Database
11: Deployment & Images	Heroku icon	Deploying to Monogo Cloud, Heroku & Cloudinary

Navigation

The screenshot shows a user interface for a learning management system. At the top, there is a header bar with various icons and links. The main content area displays a grid of course modules:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05: Models**: Incorporate Mongo Models

Navigation

Course Info

The screenshot shows a course navigation interface for 'Full Stack Web Development' taught by Eamonn de Leastar. At the top, there's a header with course details, a calendar showing 'Current Week 9', and user navigation. Below the header is a breadcrumb trail and a toolbar with various icons. The main area displays six course modules:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework.
- 03: Joi**: Validation with Joi.
- 04: TDD**: Test Driven Development.
- 05: Models**: Incorporate Mongo Models.

Navigation

Calendar



Full Stack Web Development
Eamonn de Leastar

Current Week
9

Search | Layout |

Home | Full Stack Web ... |

Overview + Assignments

Overview, schedule & assignments.

01: Node

The fundamentals of node.
A first node application.

02: Hapi.js

The Hapi application Framework

03: Joi

Validation with Joi

04: TDD

Write a failing test
Refactor
Make the test pass
Test Driven Development

05: Models

Incorporate Mongo Models

Navigation

Full Stack Development

Week No.	Type	Date Starts
1	tuition	Aug 29
2	tuition	Sep 5
3	tuition	Sep 12
Reading Week 1	reading	Sep 19
4	tuition	Sep 26
5	tuition	Oct 3
6	tuition	Oct 10
Reading Week 2	reading	Oct 17
7	tuition	Oct 24
8	tuition	Oct 31
9	tuition	Nov 7
Reading Week 3	reading	Nov 14
10	tuition	Nov 21
11	tuition	Nov 28
12	tuition	Dec 5
Study Week	Study	Dec 12
Assessments	Assessments	Dec 19

Current Week
9

Calendar

Search

Layout

Profile

More

02: Hapi.js

The Hapi application Framework

03: Joi

Validation with Joi

04: TDD

Test Driven Development

05: Models

Incorporate Mongo Models

Navigation

Search Button

The screenshot shows a course management interface for 'Full Stack Web Development' by Eamonn de Leastar. At the top, there's a navigation bar with icons for home, course, and search. The search bar is highlighted with a large black arrow pointing down to it. The main content area displays six course modules: 'Overview + Assignments', '01: Node', '02: Hapi.js', '03: Joi', '04: TDD', and '05: Models'. Each module has a thumbnail image and a brief description.

Full Stack Web Development
Eamonn de Leastar

Current Week
9 | Calendar

Search | Layout | User Profile

Overview + Assignments
The fundamentals of node.
A first node application.

01: Node

02: Hapi.js
The Hapi application Framework

03: Joi
Validation with Joi

04: TDD
Test Driven Development

05: Models
Incorporate Mongo Models

Navigation

Layout Menu

The screenshot shows a user interface for a learning platform. At the top, there's a header bar with the title "Full Stack Web Development" and the author "Eamonn de Leastar". To the right of the title are buttons for "Current Week 9" and "Calendar". Further right are "Search", "Layout" (with a dropdown arrow), and a user profile icon. Below the header is a toolbar with icons for home, back, forward, search, and other functions. The main content area displays six course modules in a grid:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05: Models**: Incorporate Mongo Models

Navigation

The screenshot displays a user interface for a "Full Stack Web Development" course by Eamonn de Leastar. The top navigation bar includes icons for home, course, search, calendar, and user profile. A sidebar on the right contains "Toggles" (Dark Mode, Compact) and "Themes" (Tutors, Dyslexia). The main content area shows a series of cards for lessons 01 through 05, each with a title, icon, and brief description.

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05:** Incorporate Mongo Models

Navigation

The screenshot shows a dark-themed user interface for a learning platform. At the top, there is a header with the title "Full Stack Web Development" and the author's name "Eamonn de Leastar". On the right side of the header are icons for search, layout, and a user profile. A sidebar on the right contains sections for "Toggles" (Dark Mode, Compact), "Themes" (Tutors, Dyslexia), and other settings.

The main area displays a grid of course cards:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05: Incorporate Mongo Models**: Incorporate Mongo Models

Each card features a small icon related to its content, such as a person icon for assignments or a database icon for MongoDB models.

Navigation

The screenshot shows a user interface for a learning management system. At the top, there is a header bar with various icons and links. The main content area displays a grid of course modules:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05: Models**: Incorporate Mongo Models

Navigation

Screenshot of a learning management system interface titled "Full Stack Web Development" by Eamonn de Leastar.

The top navigation bar includes:

- Home icon
- Course title: Full Stack Web Development
- Current Week 9
- Calendar icon
- Search icon
- Layout icon
- User profile icon (Jordan Harrison)
- More options icon

The main content area displays a grid of six course modules:

Module	Description	Icon
Overview + Assignments	Overview, schedule & assignments.	
01: Node	The fundamentals of node. A first node application.	
02: Hapi.js	The Hapi application Framework	
03: Joi	Validation with Joi	
04: TDD	Test Driven Development	
05: Models	Incorporate Mon	

The right sidebar shows the user is logged in as Jordan Harrison and includes the following options:

- Share Presence
- View 0 Online
- Tutors Time
- Github Profile
- Logout

Online Students

Full Stack Web Development
Eamonn de Leastar

Current Week
9

Calendar

Overview + Assignments

01: Node

02: Hapi.js

03: Joi

04: TDD

Client → Server

SYN message

SYN-ACK message

ACK message

10:03:28

07: Transport Layer & PT IoT

Jane Doe

John Doe

Joe Doe

Client → Server

SYN message

SYN-ACK message

ACK message

10:03:45

07: Transport Layer & PT IoT

10:03:24

07: Transport Layer & PT IoT

The image shows a learning management system interface. At the top, there's a header with a user icon, the course name 'Full Stack Web Development' by 'Eamonn de Leastar', a 'Current Week' indicator (Week 9), and a 'Calendar' button. Below the header, there are five course modules listed horizontally: 'Overview + Assignments', '01: Node', '02: Hapi.js', '03: Joi', and '04: TDD'. Each module has a small icon and a brief description. To the right of the modules, there are three vertical timelines for students: Jane Doe, John Doe, and Joe Doe. Each timeline shows a sequence of network messages (SYN, SYN-ACK, ACK) between a 'Client' and a 'Server', with specific timestamps (10:03:28, 10:03:45, 10:03:24) and session identifiers (seq=1, seq=2, seq=3). The timelines are labeled '07: Transport Layer & PT IoT'.

Navigation

Table of Contents

The screenshot shows a course management system interface for "Full Stack Web Development" by Eamonn de Leastar. The top navigation bar includes links for Home, Full Stack Web ..., Search, Layout, and a user profile icon. It also displays the current week (9) and a calendar link. Below the navigation bar is a toolbar with icons for Google Sheets, Microsoft Word, YouTube, and other tools.

The main content area displays a grid of six course modules:

- Overview + Assignments**: Overview, schedule & assignments. Includes an icon of two people at a desk.
- 01: Node**: The fundamentals of node. A first node application. Includes the Node.js logo.
- 02: Hapi.js**: The Hapi application Framework. Includes the Hapi.js logo.
- 03: Joi**: Validation with Joi. Includes the Joi logo.
- 04: TDD**: Test Driven Development. Includes a circular diagram of the TDD cycle: Write a failing test, Refactor, Make the test pass.
- 05: Models**: Incorporate Mongo Models. Includes an icon of three blue 3D cubes.

Navigation

The screenshot shows a learning platform interface with a dark theme. At the top left, there's a circular icon with a question mark and the text "Full Stack Web Development" and "Eamonn de Leastar". Below the header, there's a breadcrumb trail: a home icon, a right arrow, and the text "Full Stack Web ...". The main content area displays a grid of five course modules:

- Overview + Assignments**: Shows a dashboard with a person icon and a checklist.
- 01: Node**: Features the Node.js logo and the text "The fundamentals of node. A first node application."
- 02: Hapi.js**: Features the Hapi.js logo and the text "The Hapi application Framework"
- 03: Joi**: Features the Joi logo and the text "Validation with Joi"
- 04: TDD**: Features the TDD logo and the text "Test Driven Development"

To the right of the grid, there's a sidebar titled "Overview + Assignments" with a close button (X). The sidebar lists the course modules with expandable sections:

- 01: Node**
- 02: Hapi.js**
- 03: Joi**
- 04: TDD** (highlighted with a light green bar)
 - Test Driven Development
 - Test Driven Development
 - TDD Introduction
 - First Tests
 - TDD in Playtime
 - Joi
 - JSON Store

Navigation

The screenshot shows a user interface for a learning management system. At the top, there is a header bar with various icons and links. The main content area displays a grid of course modules:

- Overview + Assignments**: Overview, schedule & assignments.
- 01: Node**: The fundamentals of node. A first node application.
- 02: Hapi.js**: The Hapi application Framework
- 03: Joi**: Validation with Joi
- 04: TDD**: Test Driven Development
- 05: Models**: Incorporate Mongo Models

Navigation

The screenshot displays a course management system interface for "Full Stack Web Development" by Eamonn de Leastar. The top navigation bar includes links for "Course Info", "Calendar", "Full Text Search", and "Table of Contents". The main content area shows a timeline of lessons: "Overview + Assignments", "01: Node", "02: Hapi.js", "03: Joi", "04: TDD", and "05: Models". Each lesson card features a title, icon, and brief description. The left sidebar contains "Breadcrumb Navigation" and the right sidebar contains "Wall". The top right corner shows user profile information.

Course Info

Calendar

Full Text Search

Table of Contents

Breadcrumb Navigation

Wall

Layout & Themes

Search

Layout

Profile

Home

Full Stack Web ...

Overview + Assignments

01: Node

02: Hapi.js

03: Joi

04: TDD

05: Models

Companions

Overview, schedule & assignments.

The fundmendals of node. A first node application.

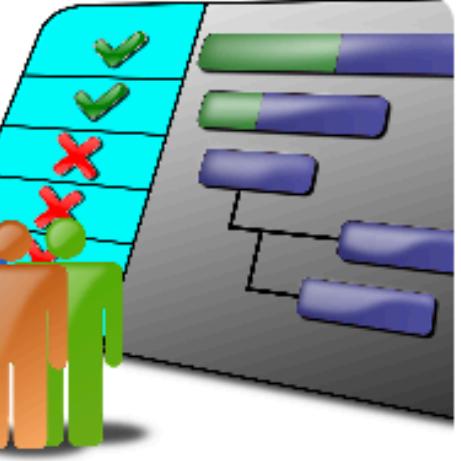
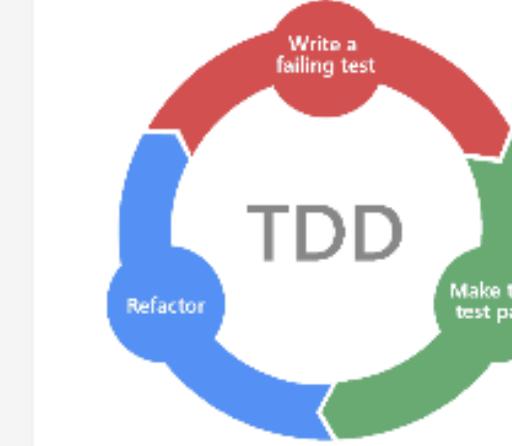
The Hapi application Framework

Validation with Joi

Test Driven Development

Incorporate Mongo Models

Topics - 1 topic/card for each week

Overview + Assignments  Overview, schedule & assignments.	01: Node  The fundamentals of node. A first node application.	02: Hapi.js  The Hapi application Framework	03: Joi  Validation with Joi	04: TDD  Test Driven Development	05: Models  Incorporate Mongo Models
06: APIs  Expose an API from an application	07: REST  Resful API implementations	08: OpenAPI  Documentating OpenAPI/Swagger APIs	09 JWT  Authenticating APIs with JSON Web Tokens	10: Seeding  Seeding the Database	11: Deployment & Images  Deploying to Monogo Cloud, Heroku & Cloudfinary

Single Topic

13: Svelte Introduction
Full Stack Web Development

Watch on YouTube

Introducing Svelte

Svelte Introduction (+ AirBnB)

- Svelte Introduction
- AirBnB Style Guide
- Airbnb Style Guide 3

A review of the fundamental features of the Svelte framework.

Javascript Style Guide from AirBnB

Blocks, Statements, Functions & Modules. This is a subset of the Airbnb...

Todo in Svelte

- Svelte First Steps
- Modules
- Lab-13-Todo 1

Building your first Svelte Application

Using npm modules in Svelte

Create a first Svelte app.

Powered by Tutors Course Reader 7.0.0
Last commit 0f29cd3

Search | Layout | 0

Watchlist | Share

SE TU netlify

Topic Video

Pre-recorded or live stream

Sub Topic 1

Sub Topic 2

Sidebar Topic

The screenshot shows a web browser window with the following details:

- Title Bar:** 19: SvelteKit Routing - Full Stack Web Development
- Page Content:** A YouTube video player for a video titled "Full Stack Web Development". The video thumbnail features a blue dashboard icon. The title "full-stack-1-19" is at the top left, and "Higher Diploma in Computer Science" is at the top right. Below the title is the text "Full Stack Web Development" with a large red play button icon. Logos for Springboard, Waterford Institute of Technology, and European Union are visible at the bottom.
- Bottom Navigation:** Buttons for "Watch on YouTube" and "Watch Later".
- Top Right:** A navigation bar with icons for Home, Search, Layout, and a user profile.



Routing & Authentication

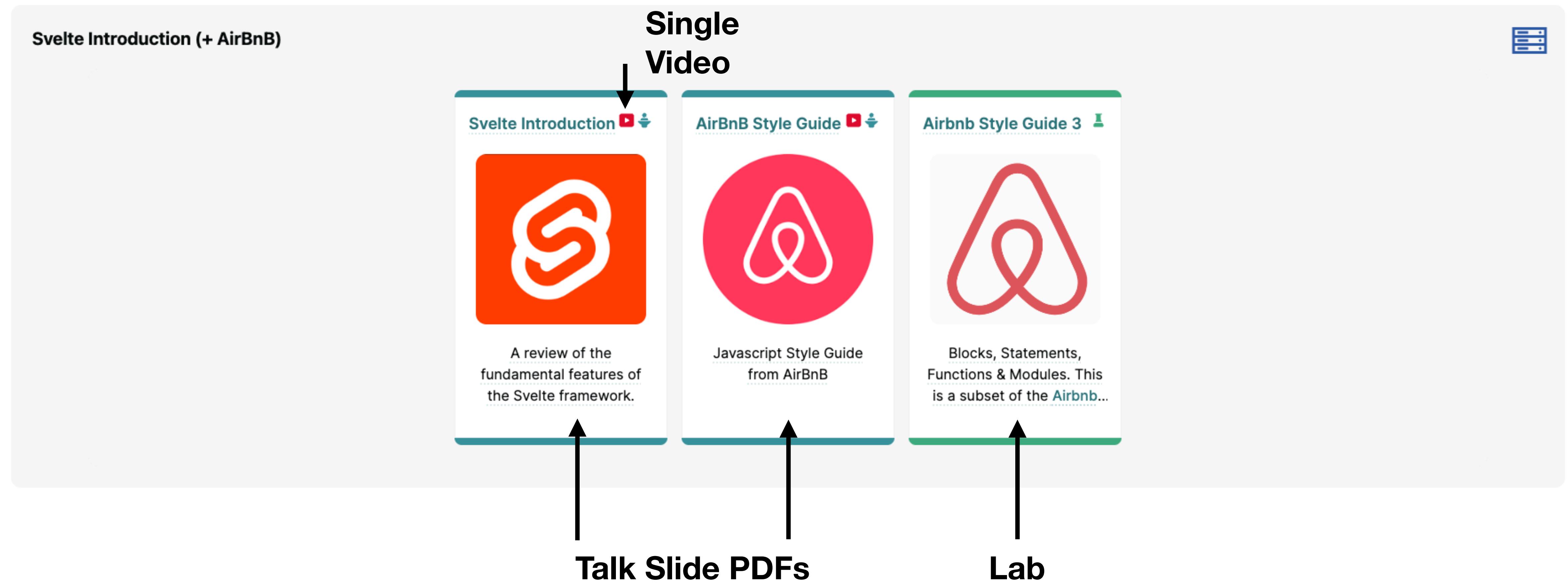
The grid contains three cards:

- Routing**: A card for "SvelteKit" showing a 3D model of a mechanical gear-like structure with a glowing orange center. The text below reads: "Introducing SvelteKit - a meta-framework for Svelte".
- Routing**: A card for "Routing" featuring a flowchart titled "HOW ROUTING WORKS". It shows the process: Incoming Request → Parse URL → Find Matching Route. If "No", it results in an "HTTP 404 Error". If "Yes", it goes to "Route Passed" and then "Process Request".
- Pages & Components**: A card showing various colorful plastic building blocks (lego-like) representing the structure of a SvelteKit application. The text below reads: "The structure of a SvelteKit application".

The sidebar is titled "Labs" and lists two sections:

- Lab-19a-donation-svelte-01-shell**: A teal box containing the text "donation-svelte-01 shell". Description: "Front end for Donation - Signup + Login routes".
- Lab-19b-donation-svelte-02-donate**: A teal box containing the text "donation-svelte-02 donate". Description: "Implement Donation + Report routes".

Sub Topic

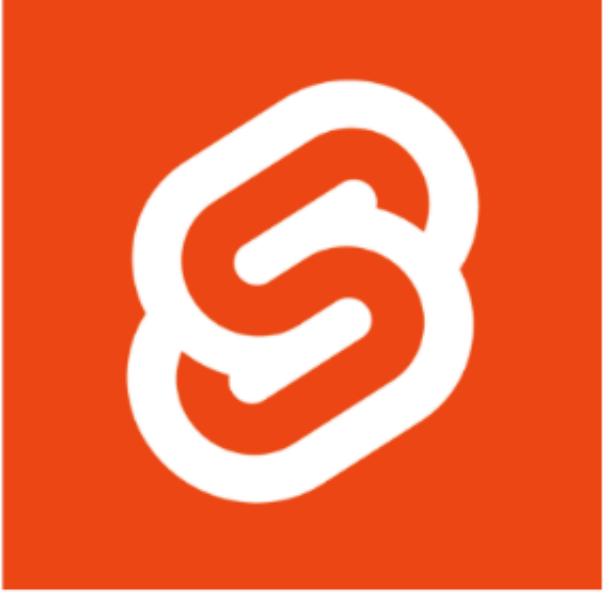


Talk View

1 of 22

Slide Navigator →

Svelte Introduction



A review of the fundamental features of the Svelte framework.

1

PDF of slides →

13: Svelte Introduction

- ▶ Introducing Svelte
- Svelte Introduction (+ AirBnB)
 - ▶ Svelte Introduction
 - ▶ AirBnB Style Guide
 - ▶ Airbnb Style Guide 3
- Todo in Svelte
 - ▶ Svelte First Steps
 - ▶ Modules
 - ▶ Lab-13-Todo 1

Topic Navigator ↑

Labs

The screenshot shows a web application interface for a "Full Stack Web Development" course. The top navigation bar includes a logo, the title "Lab-13-Todo 1", a search bar, a layout dropdown, and a user profile icon. The sidebar on the left lists various lab steps: Objectives, Create App, Style, Todo Form (which is selected and highlighted in blue), Todo List, Todo Object, Done List, npm Modules, and Exercises. A large arrow points upwards from the "Lab Steps Navigator" label to the sidebar. The main content area features a heading "Todo Form" and instructions to include it in "App.svelte". Below this is a code block for "App.svelte" containing Svelte code for a todo form. A second arrow points to the right from the "Lab Step Content" label, which is positioned next to the "Simple Todo List" screenshot.

Lab Steps Navigator

Lab Step Content

Todo Form

Include this form in App.svelte, just below the header:

App.svelte

```
<div class="section box">
  <div class="field is-horizontal">
    <div class="field-label is-normal">
      <label for="todo" class="label">What should I do?</label>
    </div>
    <div class="field-body">
      <div class="field">
        <p class="control">
          <input id="todo" class="input" type="text" placeholder="Type something...">
        </p>
        <button class="button">Add Todo</button>
      </div>
    </div>
  </div>
</div>
```

Just by saving the file the UX should automatically update:

The screenshot shows a browser window with a tab labeled "Todo using Svelte". The address bar shows "localhost:3000". The page title is "Simple Todo List" with the subtitle "Fun things to do". At the bottom, there is an input field with the placeholder "What should I do?" and a button labeled "Add Todo".



Objectives

Text

Tables, Lines and Images

Links and Code Blocks

Images

Latex

Reference

Exercises & Archives

Latex

Latex support via the [Katex Library](#).

Example 0

$$E = mc^2$$

Example 1

This is an inline example: $c = \pm\sqrt{a^2 + b^2}$ with text before and after

Example 2

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Example 3

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

Example 4

$$g(x) = 3x^4 + 2x^2$$

Example 5

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \sum_{i=1}^n i^3 = \left(\frac{n(g(n)+1)}{2} \right)^2 c = \pm\sqrt{a^2 + b^2}$$

Example 6

$$\frac{1}{\left(\sqrt{\phi\sqrt{5}-\phi}\right)e^{\frac{2}{5}\pi}} = 1 + \frac{e^{-2\pi}}{1 + \frac{e^{-4\pi}}{1 + \frac{e^{-6\pi}}{1 + \frac{e^{-8\pi}}{1 + \dots}}}}$$

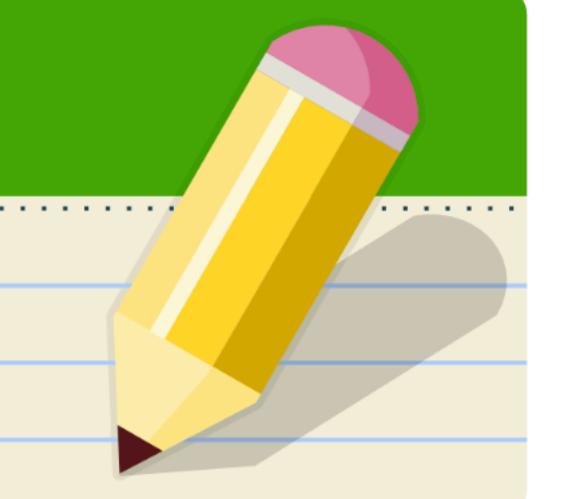
Example 7

$$1 + \frac{q^2}{(1-q)} + \frac{q^6}{(1-q)(1-q^2)} + \dots = \prod_{j=0}^{\infty} \frac{1}{(1-q^{5j+2})(1-q^{5j+3})}, \quad \text{for } |q| < 1.$$

Latex Lab Content

Note

Note 1



This is an example of a note.

Note 1

This is an example of a note.

It can have images:



A code block with syntax highlighting:

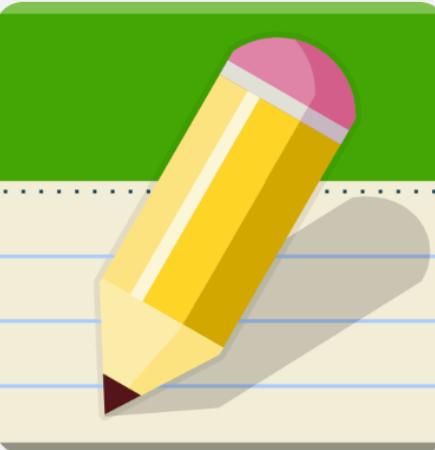
user-api-test.js

```
import { playtimeService } from "./playtime-service.js";

suite("User API tests", () => {
  setup(async () => {
  });
  teardown(async () => {
  });

  test("create a user", async () => {
  });
});
```

Simple



Main Lesson

- Lecture 1
- Lecture 2
- Lab-01

Supporting Resources

- Note 1
- Web Site 1
- Resource 1

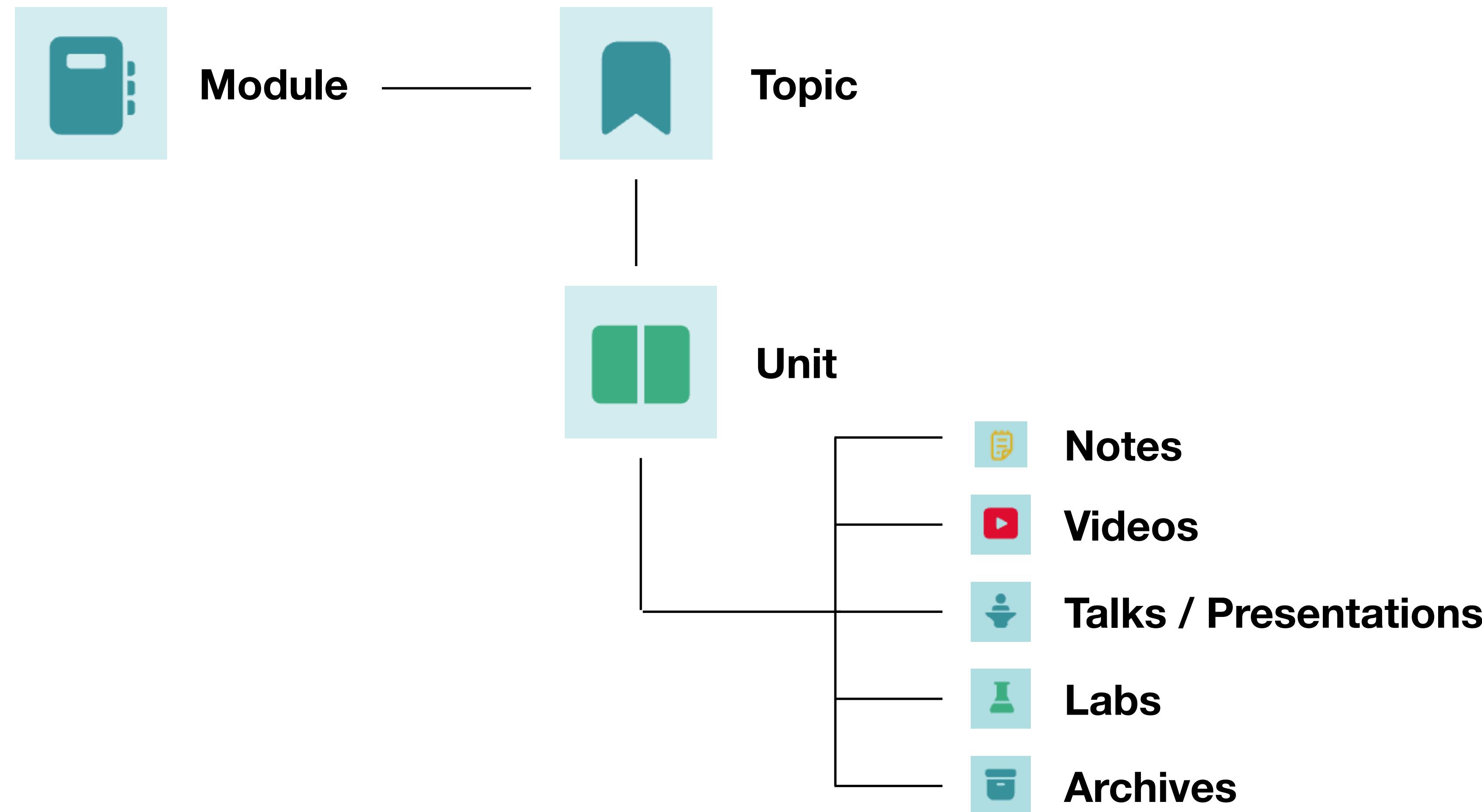
Educator Experience

Educator Experience



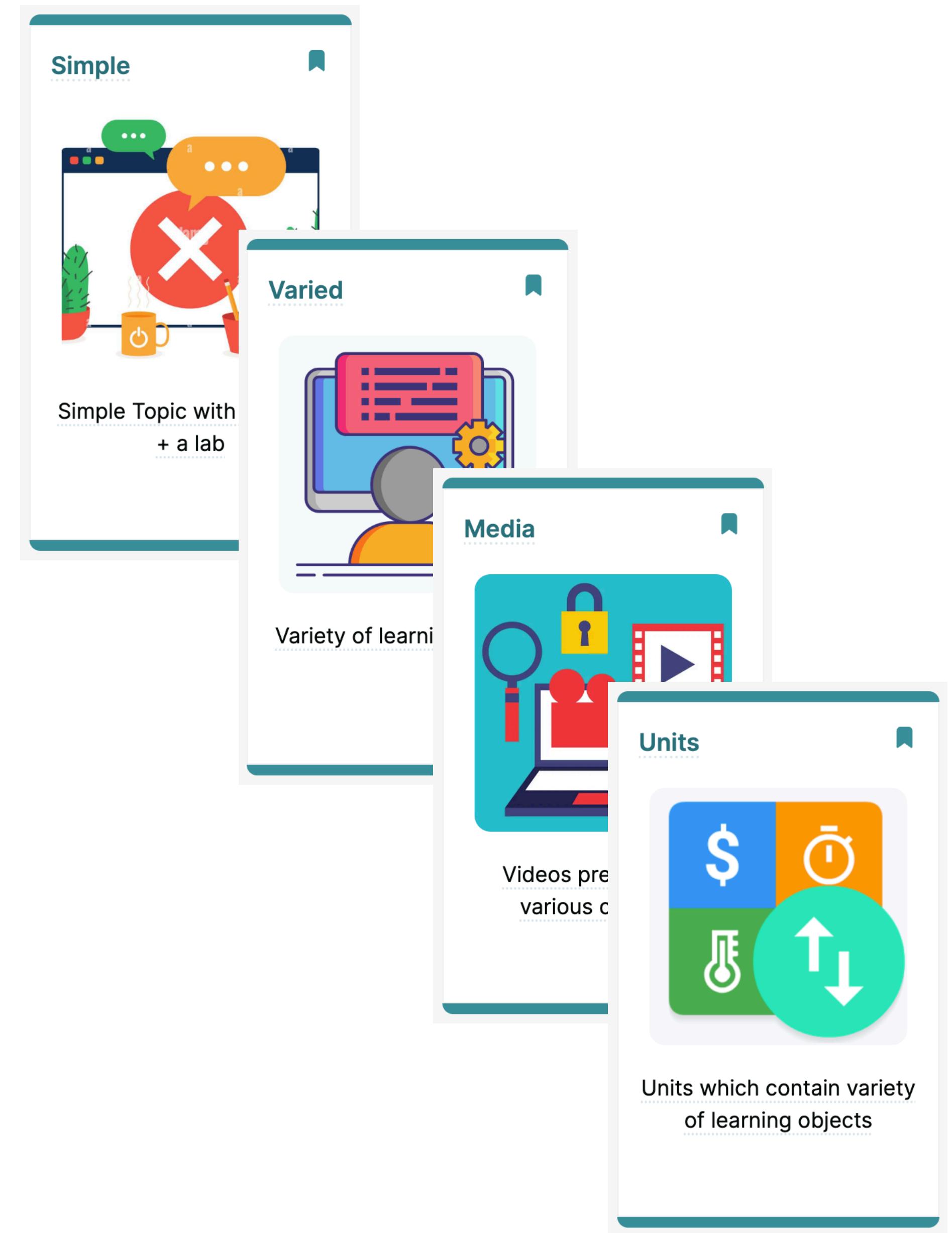
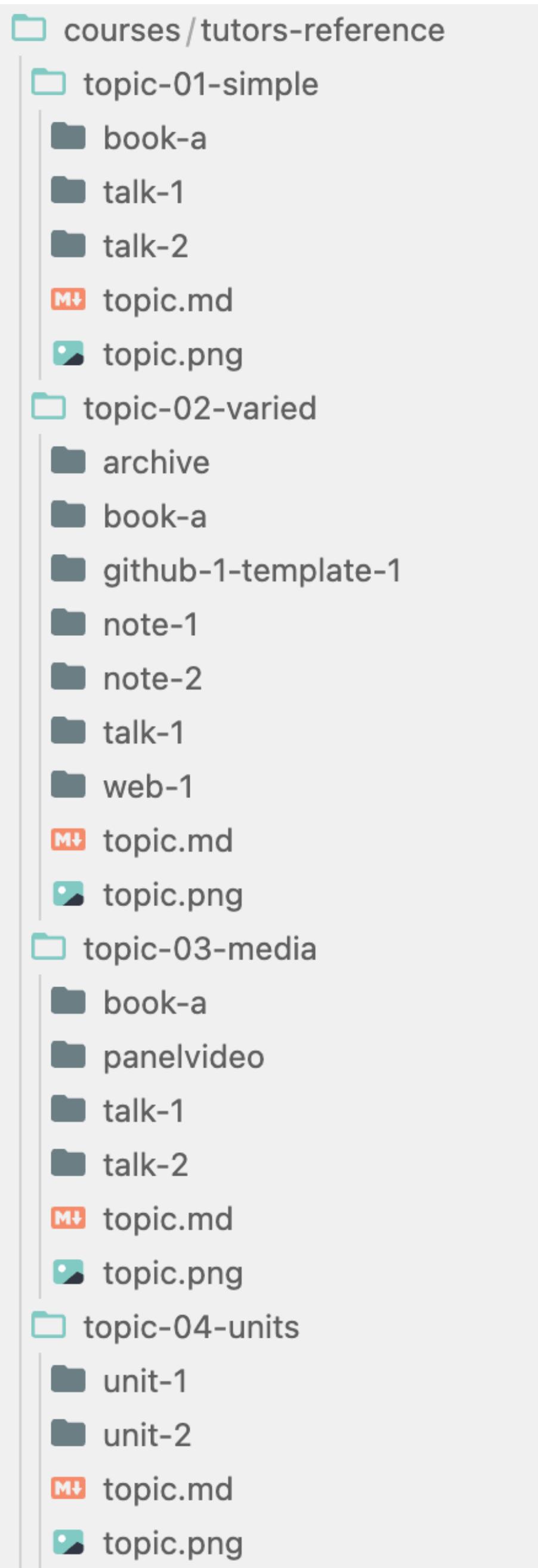
The ***Educator Experience*** prioritises the creation of a **guided paths** through a curriculum via the creation of learning materials that are **autonomous, structurally aligned, composable, auditable, extensible, versioned** and **independent**.

Module Structure

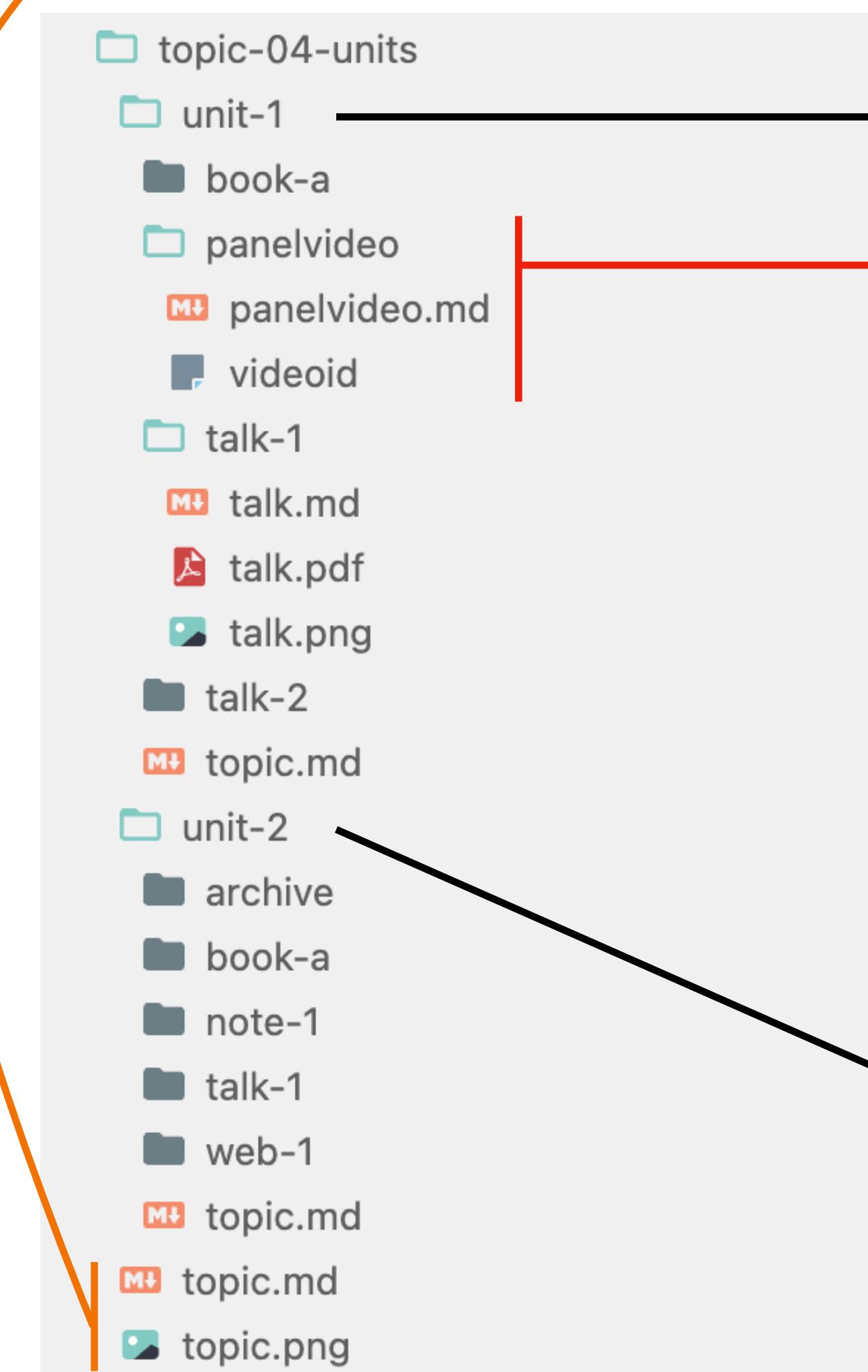
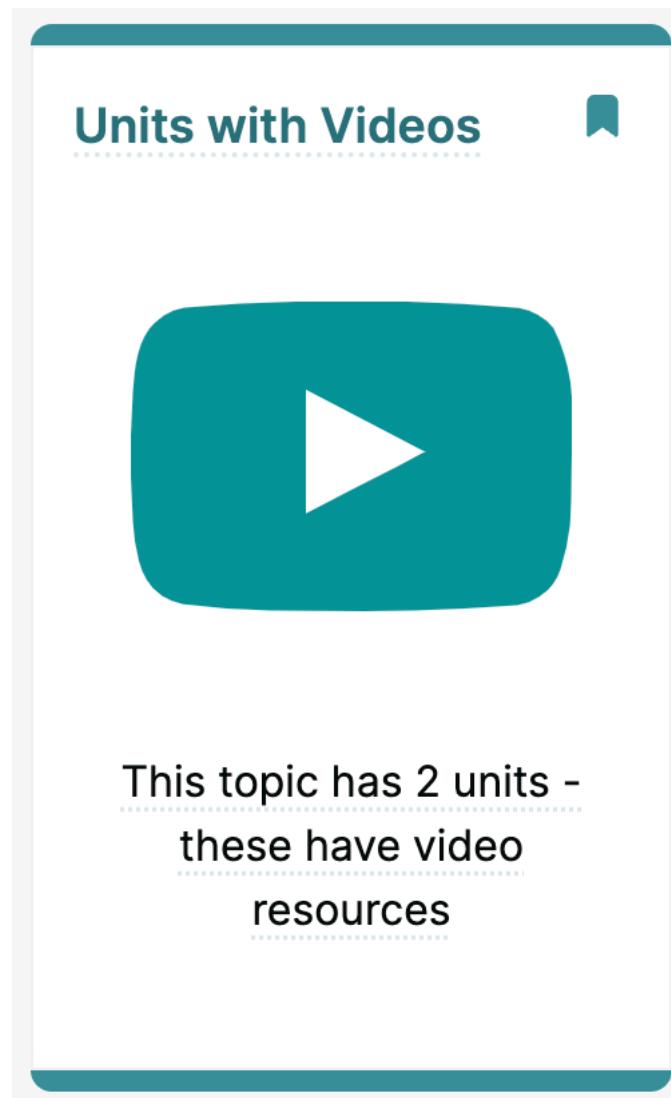


Topics Structure

<https://reference.tutors.dev>



Single Topic Structure



A screenshot of a course reader application showing the topic structure:

Unit 1 Title

Higher Diploma in Computer Science
Full Stack Web Development

Watch later Share

Panel Video

Lecture

A short summary of the talk, no more than two sentences. Avoid bullet...

Lecture

Provide a short summary, perhaps supported by a representative image.

Lab-01

A set of practical instructions or a walk-through

Note Example

This is an example of a note

Resource I

This is not a pdf link, but a zipped archive. If it is selected the archive will ...

Web Site

A web site of interest

Lecture

A short summary of the talk, no more than two sentences. Avoid bullet...

Lab-01

A set of practical instructions or a walk-through

Powered by Tutors Course Reader 7.0.0
Last commit 0f29cd3

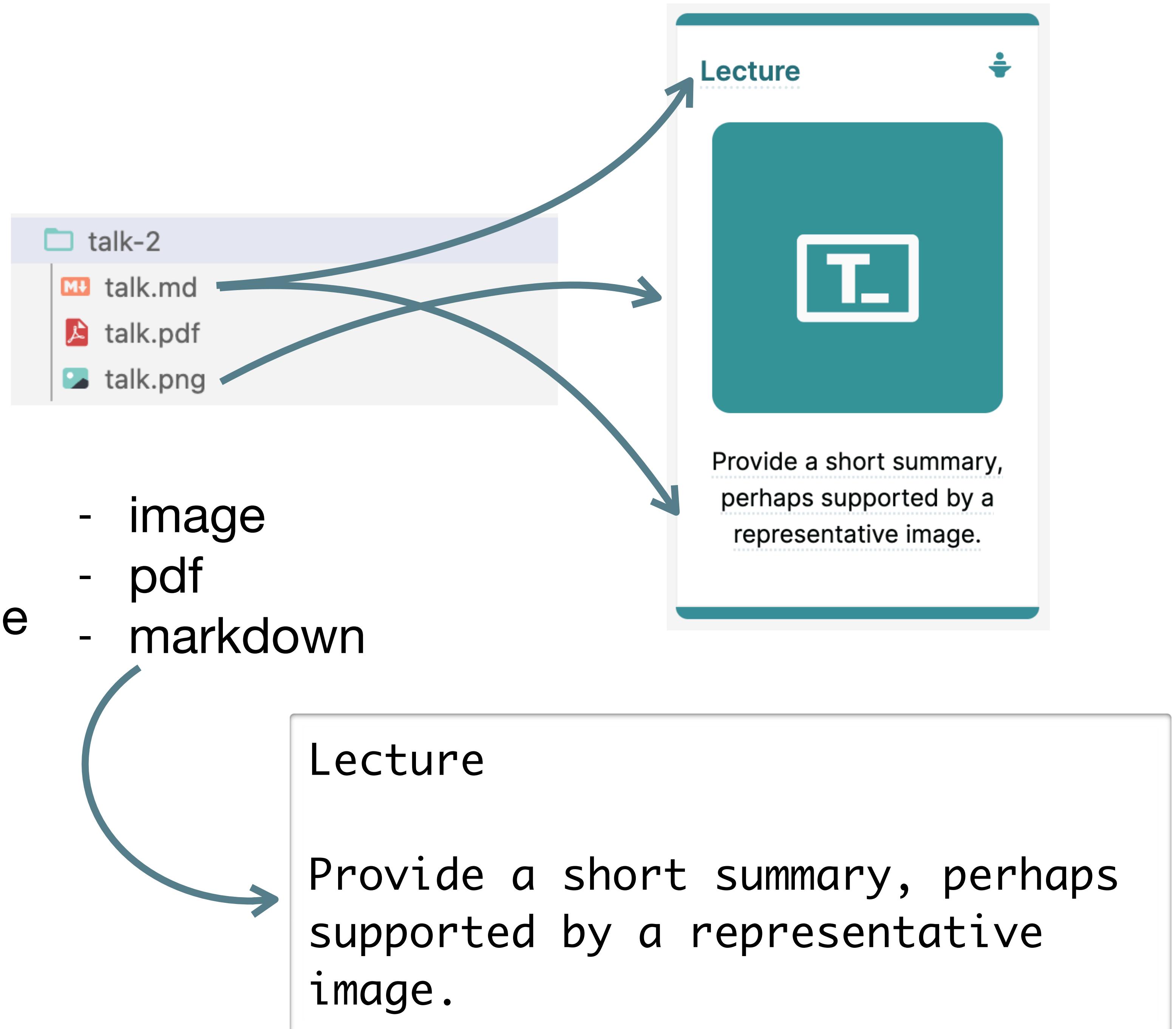
SE TU

REPLIES BY netlify

Talk Structure

all files
same name

- image
- pdf
- markdown



Lab Structure

```
book-a
archives
archive.zip
img
01.jpg
02.jpg
03.png
04.png
main.png
00.Lab-01.md
01.01.md
02.02.md
03.03.md
04.04.md
05.05.md
06.Exercises.md
```

Lab-01: Objectives
01: Text
02: Tables, Lines and Images
03: Links and Code Blocks
04: Images
05: Katex
Exercises: Exercises & Archives

Links and Code Blocks

You can insert links in bullet points:

- <http://github.com>
- <http://bitbucket.dom>

More Code Block examples

```
publish(path, course) {
    const basePath = '../../../../../' + path + '/' + this.folder;
    futils.initEmptyPath(basePath);

    this.resources = this.talks.concat(this.labs);
    futils.copyFileToFolder(this.img, basePath);
    futils.writeFile(basePath + '/index.html', nunjucks.render('topic.html', this));

    futils.writeFile(basePath + '/ajaxlabel.html',
        nunjucks.render('ajaxlabel.html',
            { url: this.url.substring(5) + '/' + this.folder }));
    futils.writeFile(basePath + '/indexmoodle.html', nunjucks.render('indexmoodle.html', this));

    console.log(this.title);
    this.publishTalks(basePath);
    this.publishLabs(basePath, course);
}
```

courses > tutors-reference > topic-01-simple > book-a > 03.03.md > # Links and Code Blocks

```
1 # Links and Code Blocks
2
3 You can insert links in bullet points:
4
5 - <http://github.com>
6 - <http://bitbucket.dom>
```

More Code Block examples

```
11 ~~~
12 publish(path, course) {
13     const basePath = '../../../../../' + path + '/' + this.folder;
14     futils.initEmptyPath(basePath);

15     this.resources = this.talks.concat(this.labs);
16     futils.copyFileToFolder(this.img, basePath);
17     futils.writeFile(basePath + '/index.html', nunjucks.render('topic.html', this));

18     futils.writeFile(basePath + '/ajaxlabel.html',
19         nunjucks.render('ajaxlabel.html',
20             { url: this.url.substring(5) + '/' + this.folder }));
21     futils.writeFile(basePath + '/indexmoodle.html', nunjucks.render('indexmoodle.html', this));

22     const ~~/Documents/GitHub/tutors/courses/tutors-
23     this.reference/topic-03-media
24     this.publishLabs(basePath, course);
25
26     ~~~
27
28 }
29
30
31 This was autodetected as javascript. This one is html:
32
33 ~~~
```

Publishing

There are 2 ways to publish a module.

A dark teal rounded rectangle containing the text "Dynamic/JSON".

Dynamic/JSON

A dark teal rounded rectangle containing the text "Static/HTML".

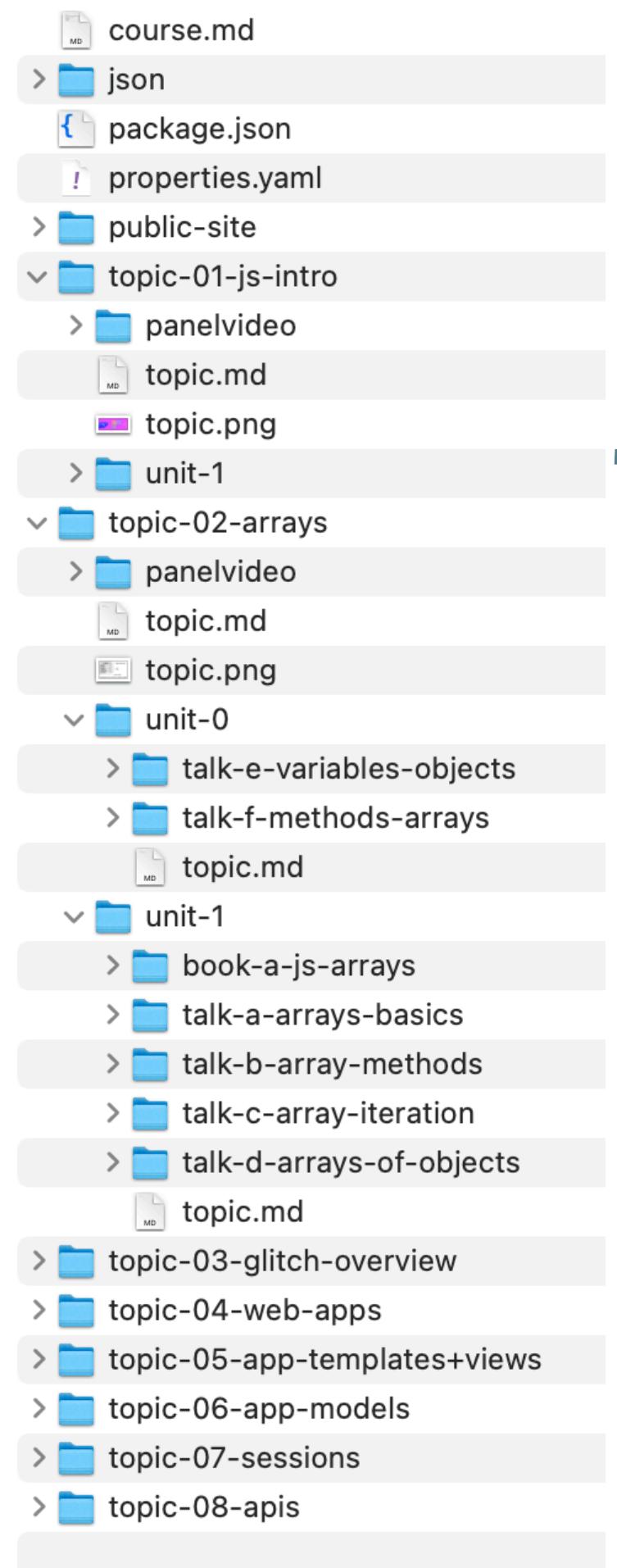
Static/HTML

Dynamic/JSON Publishing

- Dynamic Site
- Browser through the Tutors Reader app
- Continuously updated and enhanced
- Included optional analytics or presence features

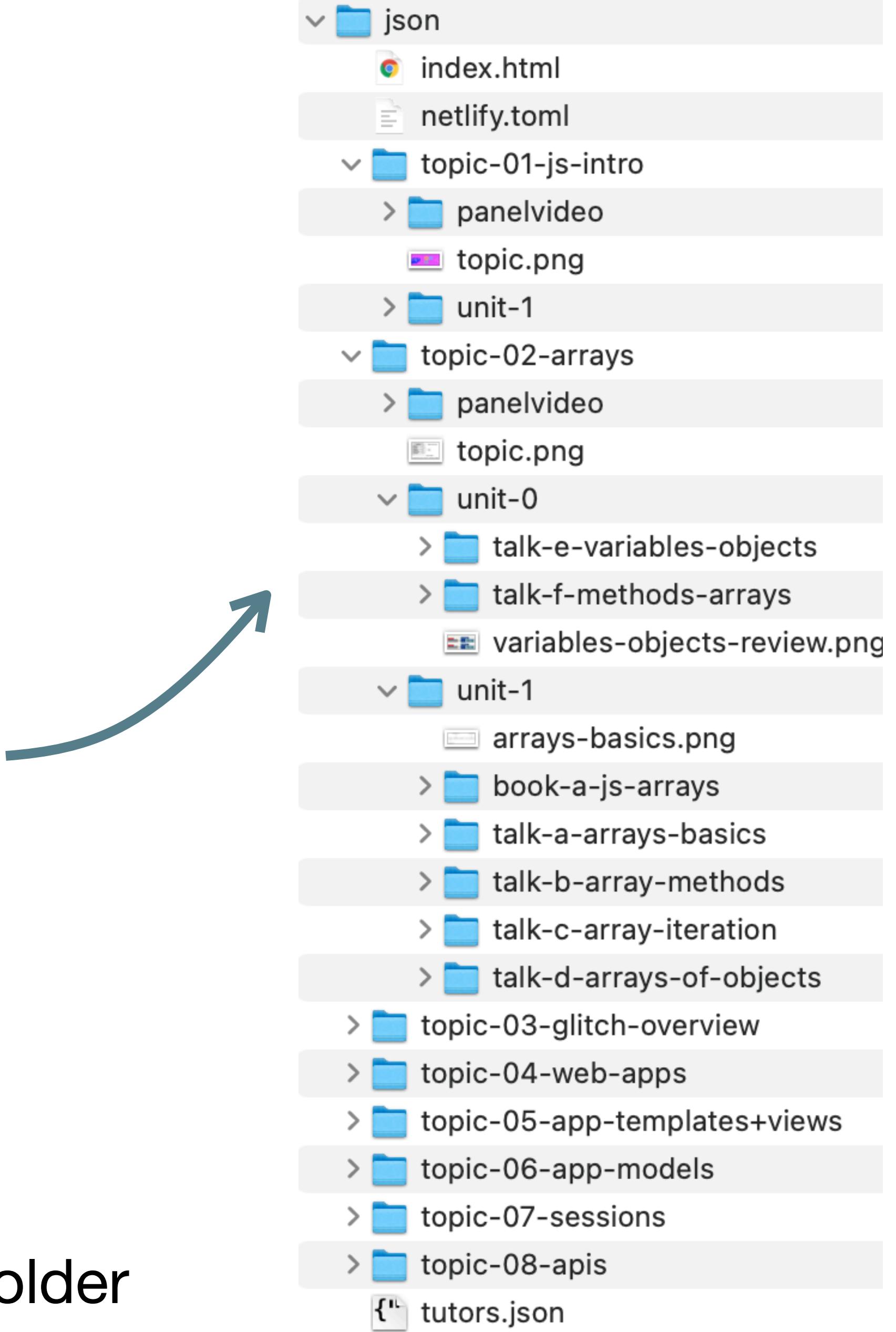


Dynamic/JSON Publishing



npx tutors-json

```
-bash
Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-json
Static course generator tutors-json 2.2.3 (tutors-lib: ^0.6.3)
::  Web Development II
--> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
--> Introducing Javascript (unit)
:: Introducing Javascript
--> Module Overview (talk)
--> JS Introduction (talk)
--> Variables & Logic (talk)
--> Const, Let & Objects (talk)
--> Lab-01 JS Intro(lab)
--> 1: Introducing Javascript (panelvideo)
--> 2: Javascript Arrays (topic)
:: 2: Javascript Arrays
--> Javascript Variables, Objects & Methods (unit)
:: Javascript Variables, Objects & Methods
--> Variables & Objects Review (talk)
--> Methods & Arrays Review (talk)
--> Javascript Arrays (unit)
:: Javascript Arrays
--> Arrays: Basics (talk)
--> Array Methods (talk)
--> Array Iteration (talk)
--> Arrays of Objects (talk)
--> Lab-02 JS Arrays(lab)
--> 02: Javascript Arrays (panelvideo)
--> 3: Glitch Overview (topic)
:: 3: Glitch Overview
--> Introducing Glitch (unit)
```



Command generates static site to 'json' folder

Dynamic/JSON Publishing

The diagram illustrates the workflow for publishing dynamic content:

- Course Structure:** On the left, a file tree shows a folder named "json" containing "index.html", "netlify.toml", and several "topic-X" sub-folders. Each "topic-X" folder contains files like "panelvideo", "topic.png", and "unit-Y".
- Netlify Deployment:** In the middle, the Netlify dashboard shows a "Tutors Starter" team. A large arrow points from the "json" folder to the dashboard, indicating the source of the deployment.
- Reader Experience:** On the right, a screenshot of a web browser shows a course page titled "Web Development II" by Eamonn de Leastar, WIT Computing. The page displays a grid of 14 modules, each with a title, description, and icon. Modules include "1: Introducing Javascript", "2a: Javascript Arrays", "3: Web Applications", "4: Templates & Routes", "5: Forms", "6: Sessions", and "7: APIs".

The screenshot shows the Netlify site overview for "tender-hamilton-28ce6a". Key details include:

- Site Overview:** Shows bandwidth used (0 KB/100 GB), build minutes used (0/300), concurrent builds (0/1), and members (1).
- Getting started:** Three steps are outlined:
 - 1 Your site is deployed ✓ Try a test build and deploy, directly from your Git repository or a folder.
 - 2 Set up a custom domain → Buy a new domain or setup a domain you already own.
 - 3 Secure your site with HTTPS Your site is secured automatically with a Let's Encrypt certificate.
- Production:** Status is "Published" at 11:22 AM with no deploy message.

1. Netlify GitHub deployment
2. Netlify Drag & Drop

Deploys the Course to be read by the tutors reader

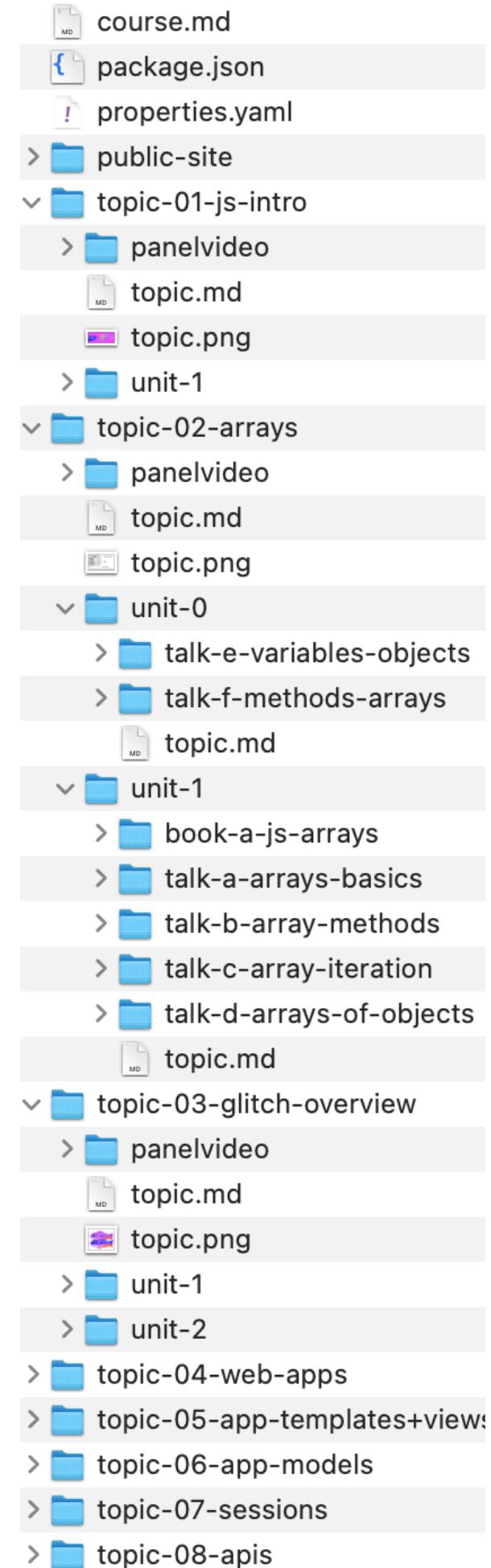
Static/HTML Publishing



Static/HTML

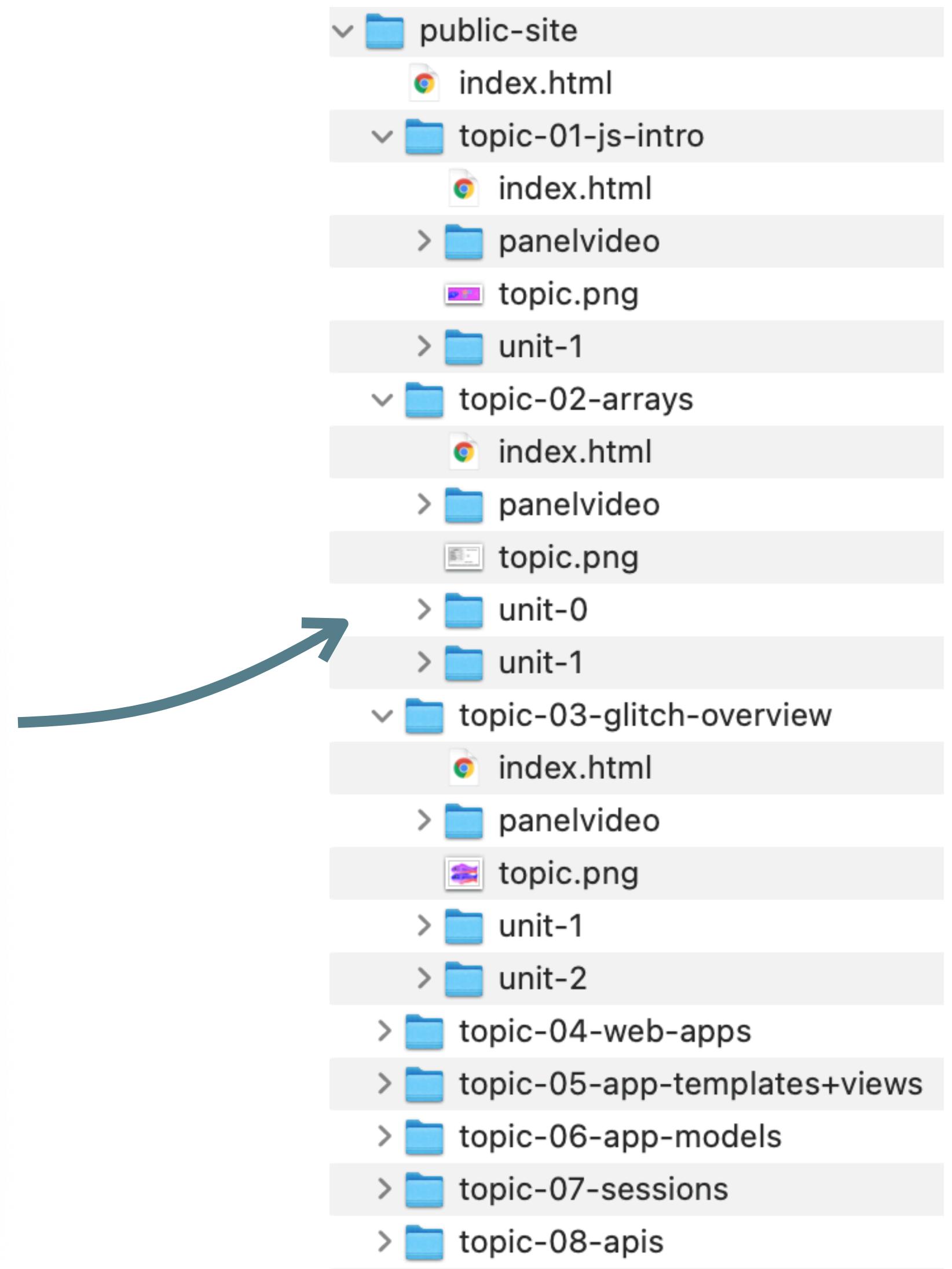
- Static Site
- Can be browsed offline (distribute zipped archive) / uploaded to traditional web host
- Format ‘frozen’ at time of generation
- No analytics or presence features

Static/HTML Publishing



npx tutors-html

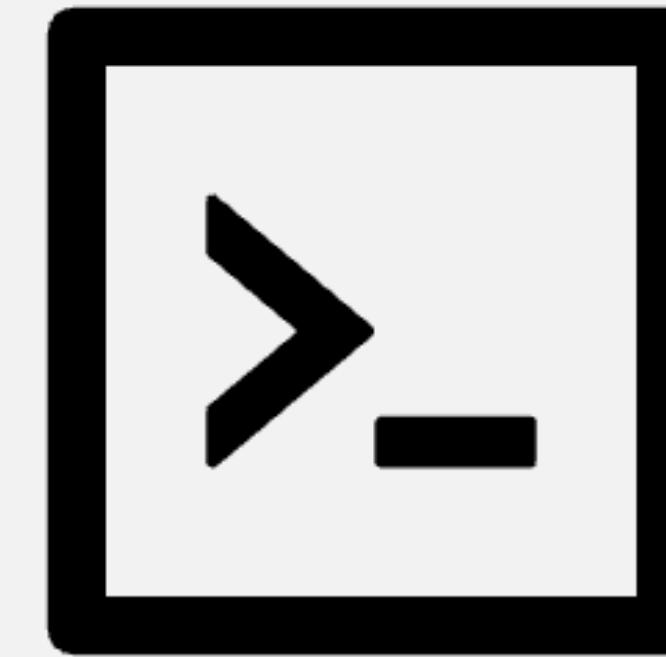
```
-bash
Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-html
Static course generator tutors-json 0.6.7 (tutors-lib: ^0.6.5)
:: Web Development II
  --> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
  --> Introducing Javascript (unit)
:: Introducing Javascript
  --> Module Overview (talk)
  --> JS Introduction (talk)
  --> Variables & Logic (talk)
  --> Const, Let & Objects (talk)
  --> Lab-01 JS Intro(lab)
  --> 1: Introducing Javascript (panelvideo)
  --> 2: Javascript Arrays (topic)
:: 2: Javascript Arrays
  --> Javascript Variables, Objects & Methods (unit)
:: Javascript Variables, Objects & Methods
  --> Variables & Objects Review (talk)
  --> Methods & Arrays Review (talk)
  --> Javascript Arrays (unit)
:: Javascript Arrays
  --> Arrays: Basics (talk)
  --> Array Methods (talk)
  --> Array Iteration (talk)
  --> Arrays of Objects (talk)
  --> Lab-02 JS Arrays(lab)
  --> 02: Javascript Arrays (panelvideo)
  --> 3: Glitch Overview (topic)
:: 3: Glitch Overview
```



Command generates static site to ‘public-site’ folder

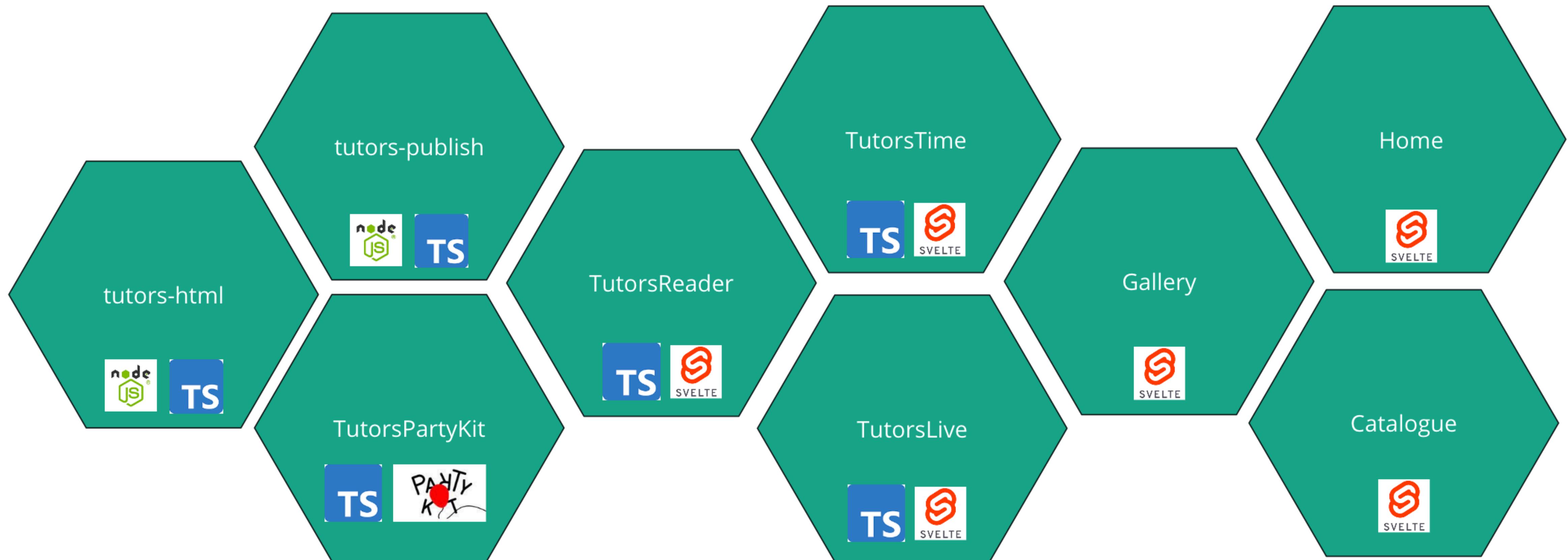
Developer Experience

Developer Experience

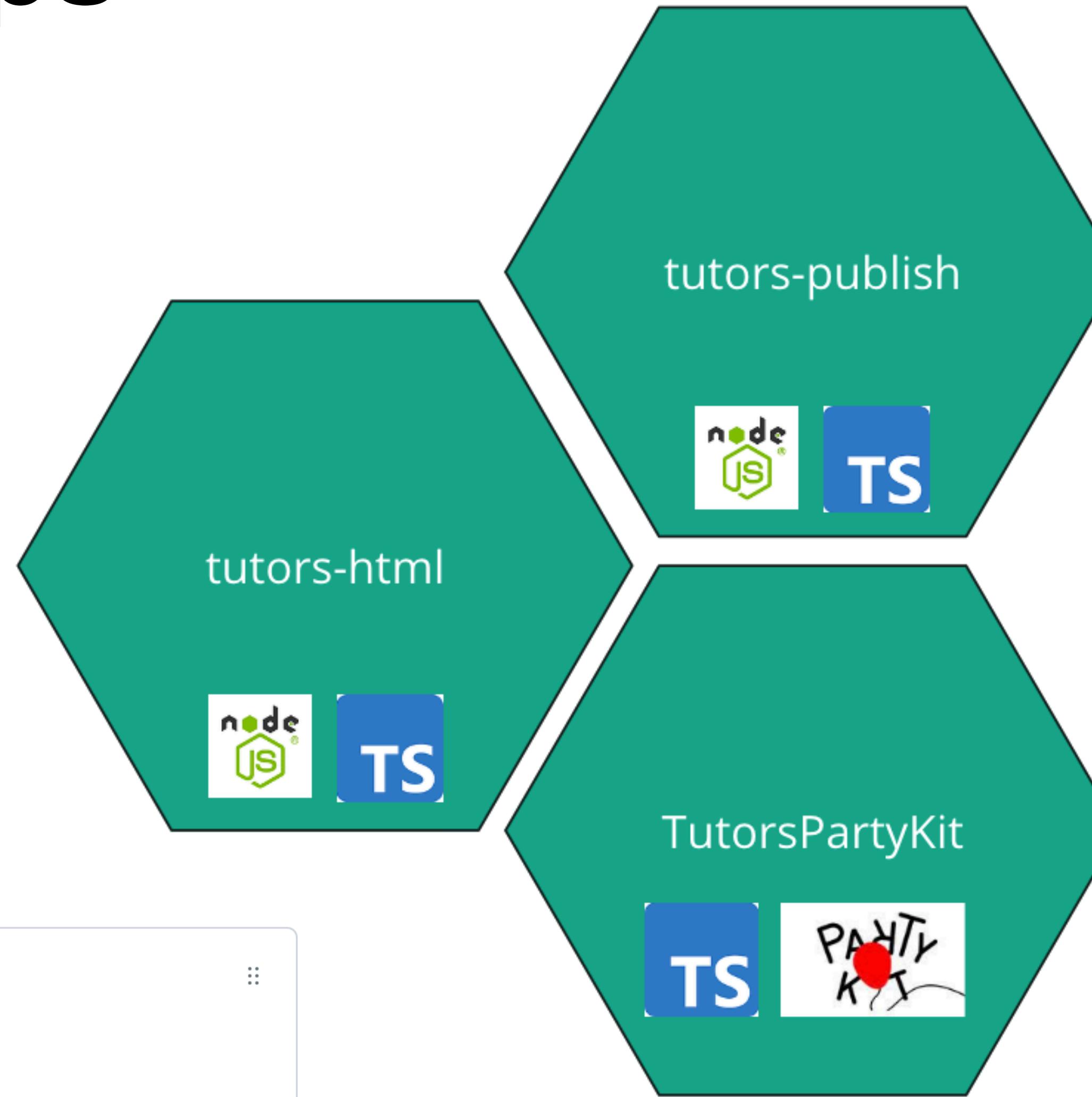
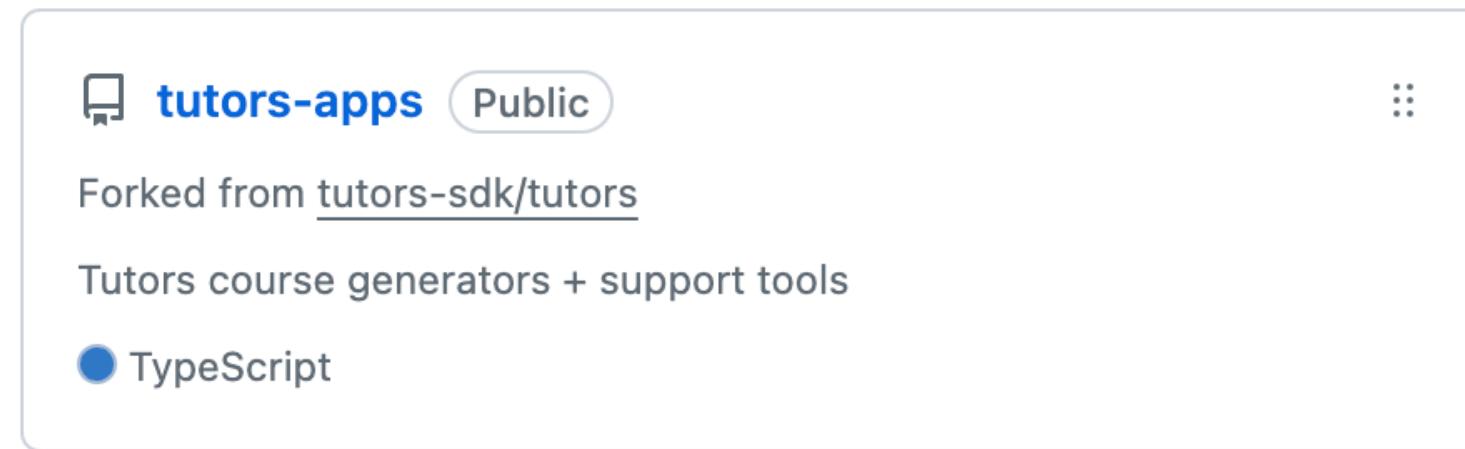


The ***Developer Experience*** prioritises the specification and implementation of **robust, well documented, loosely coupled components & services**, integrated into a **coherent toolkit** open to contributions from **diverse skill sets**.

Tutors Applications



TutorsApps



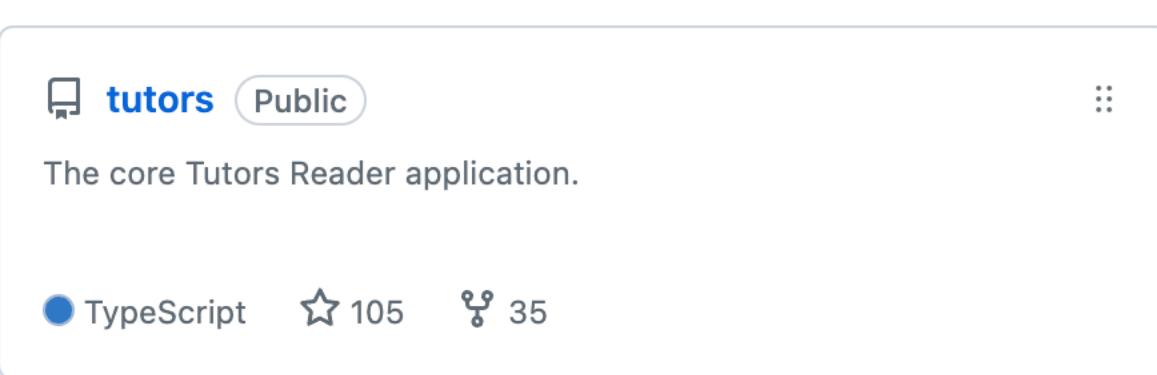
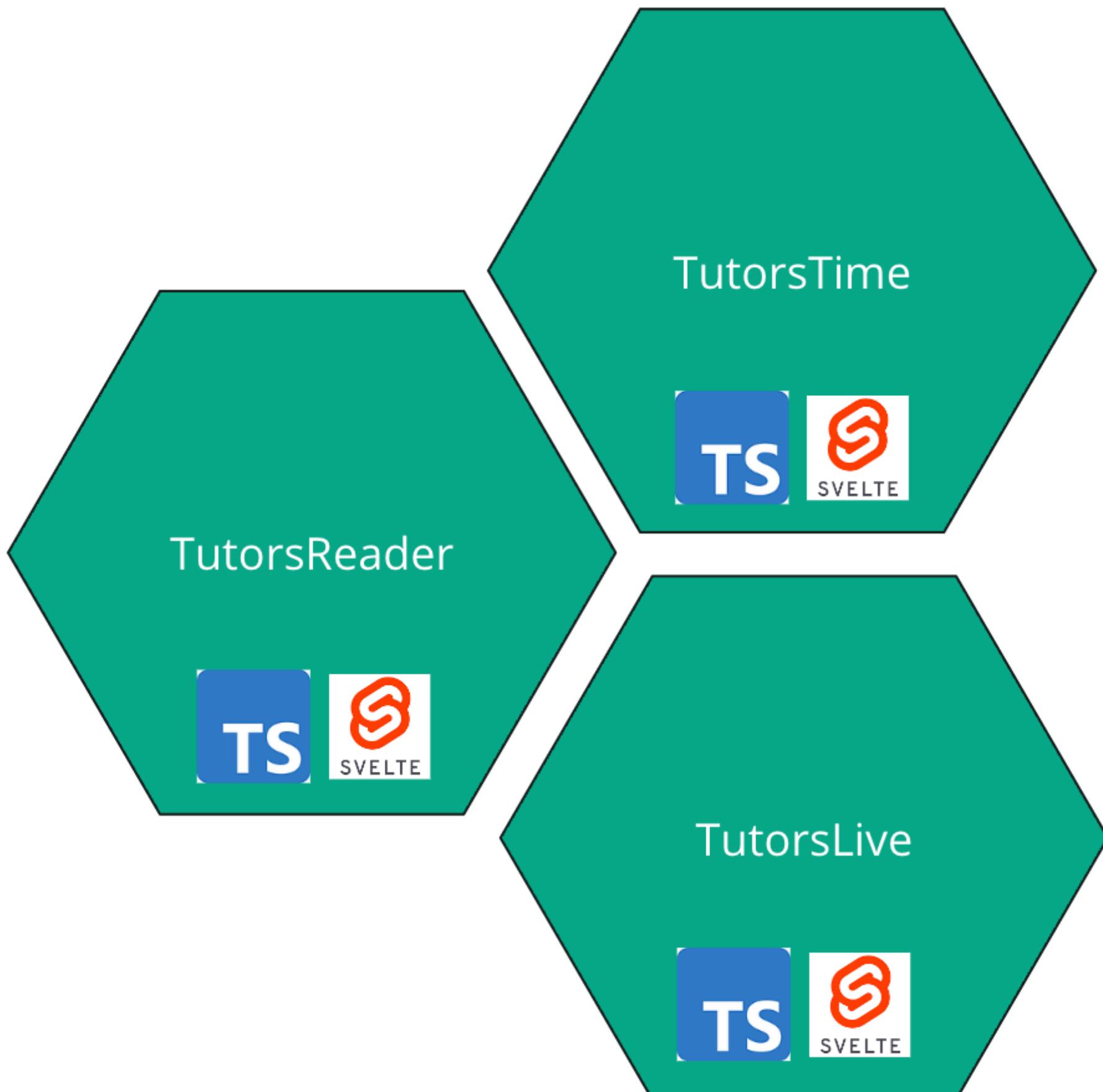
<https://github.com/tutors-sdk/tutors-apps>

A screenshot of the file tree for the 'tutors-apps' repository, showing the following structure:

- cli
 - tutors-gen-lib
 - src
 - .eslintrc.cjs
 - package.json
 - tsconfig.json
 - tutors-publish
 - src
 - .eslintrc.cjs
 - package.json
 - tsconfig.json
 - tutors-publish-html
 - src
 - .eslintrc.cjs
 - .prettierrc
 - package.json
 - tsconfig.json
 - party/tutors-party-kit
 - public
 - src
 - .gitignore
 - package.json
 - partykit.json
 - README.md
 - tsconfig.json

Tutors

```
src
  lib
    services
      models
      types
      utils
      analytics.ts
      course.ts
      firebaseAnalytics.ts
      presence.ts
      supabaseAnalytics.ts
    ui
      app-shells
      learning-objects
      navigators
      support
      themes
      time
      animations.ts
      environment.ts
      stores.ts
```



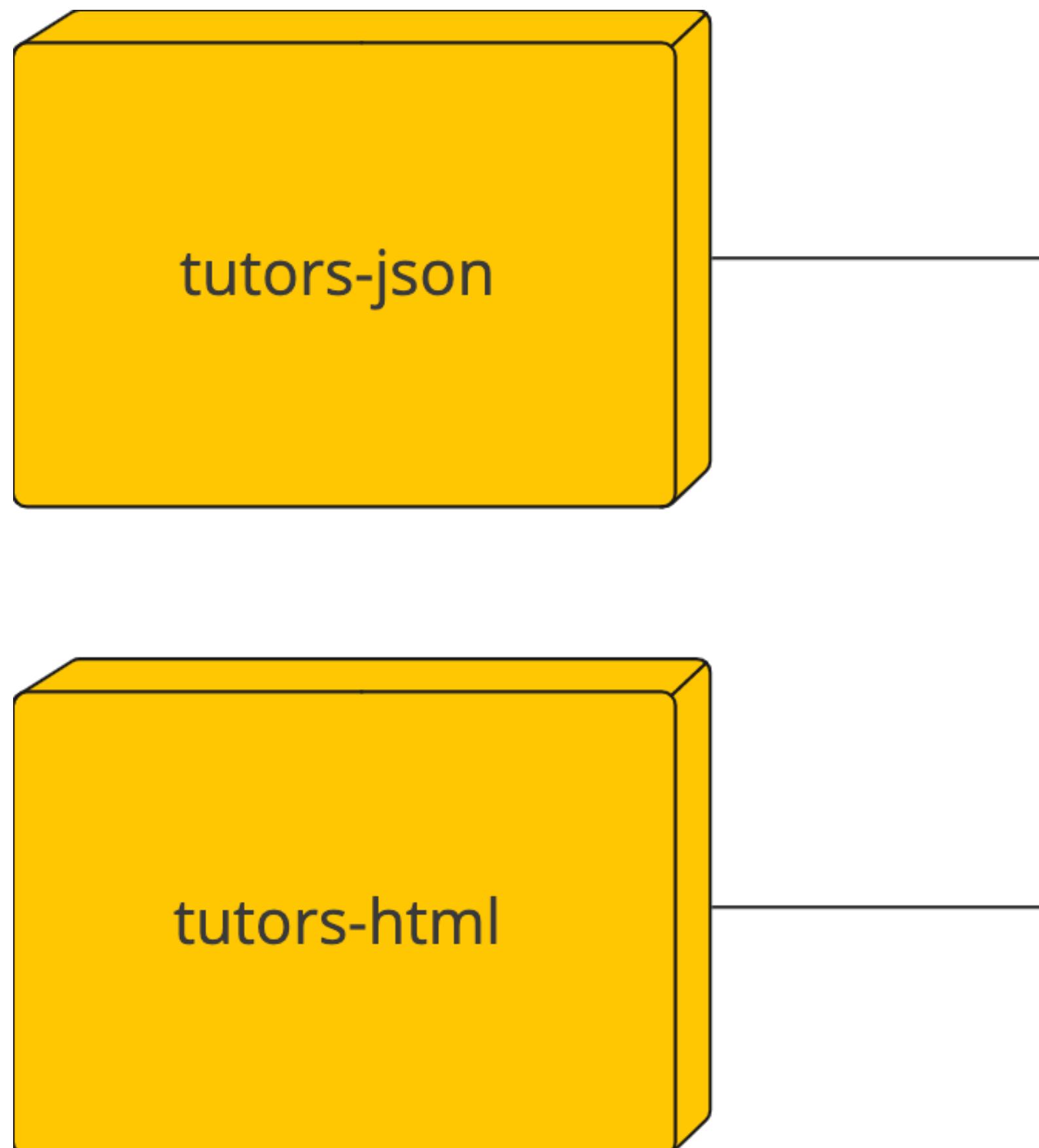
> routes
> auth
> course-reader
> course
> lab
> note
> search
> talk
> topic
> video
> wall
> +layout.svelte
> home
> (time)
> active
> catalogue
> gallery
> live
> simulate
> time
> +error.svelte
TS +layout.server.ts
> +layout.svelte
TS +layout.ts
TS app.d.ts
TS app.html
TS app.postcss
TS hooks.server.ts

<https://github.com/tutors-sdk/tutors>

Generator Architecture

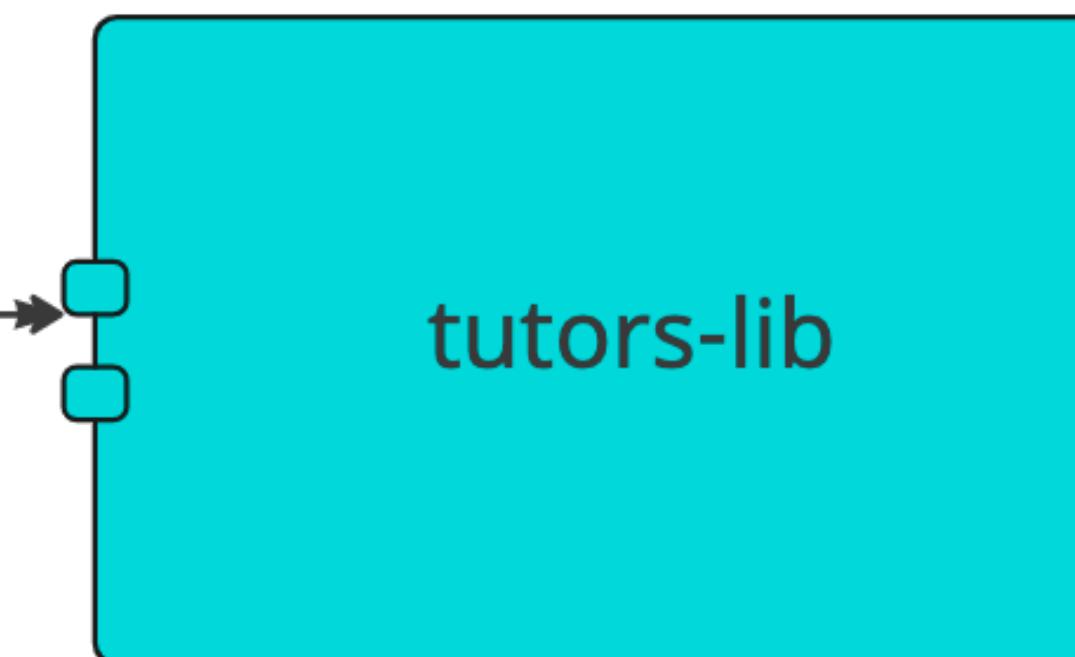
Node.js Command Line Applications

Generate JSON model for course



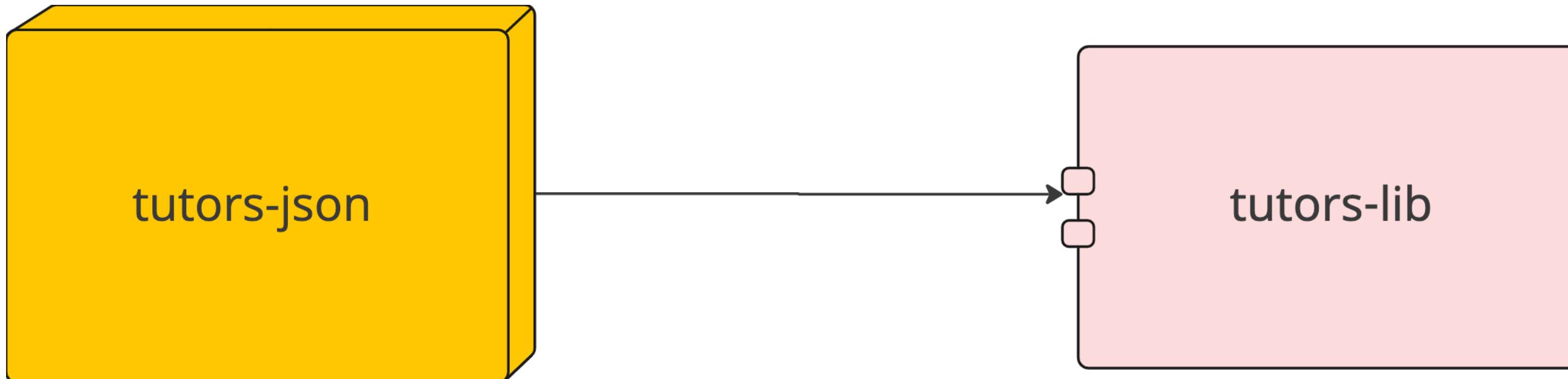
Generate HTML Static site for course

Course Memory Model



“Abstract Syntax Tree” for course folder/file structure

```
controllers
  TS json-emitter.ts
  TS netlify.ts
  TS tutors-json.ts
```



```
models
  TS course.ts
  TS lab.ts
  TS lo.ts
  TS los.ts
  TS note.ts
  TS topic.ts
  TS web-los.ts
```

```
utils
  TS futils.ts
  TS lutils.ts
  TS properties.ts
  TS videoutils.ts
```

```
course.md
> json
  package.json
  properties.yaml
> public-site
  <topic-01-js-intro
    panelvideo
    topic.md
    topic.png
  > unit-1
  <topic-02-arrays
    panelvideo
    topic.md
    topic.png
  > unit-0
    talk-e-variables-objects
    talk-f-methods-arrays
    topic.md
  <unit-1
    book-a-js-arrays
    talk-a-arrays-basics
    talk-b-array-methods
    talk-c-array-iteration
    talk-d-arrays-of-objects
    topic.md
  > topic-03-glitch-overview
  > topic-04-web-apps
  > topic-05-app-templates+views
  > topic-06-app-models
  > topic-07-sessions
  > topic-08-apis
```

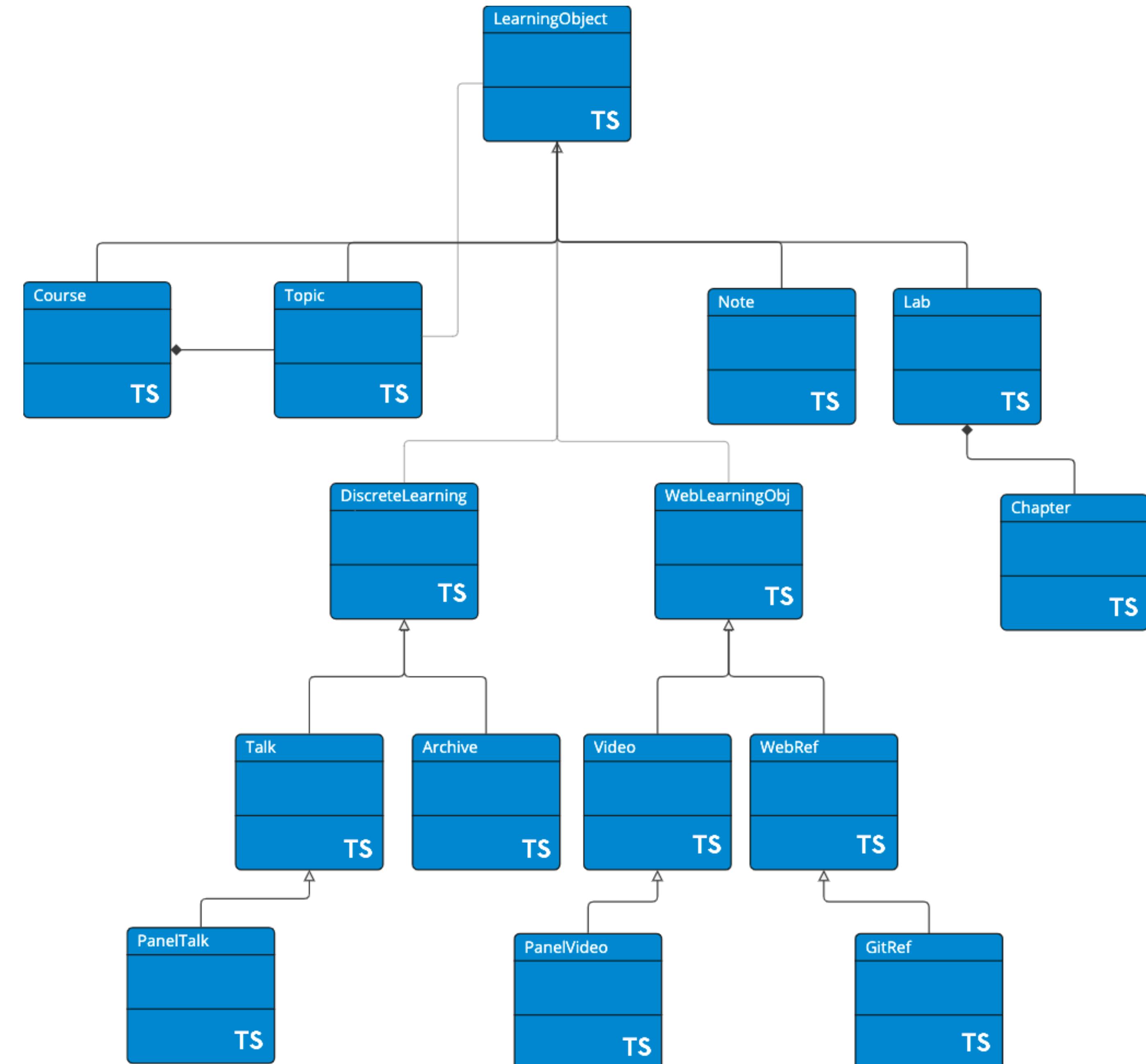
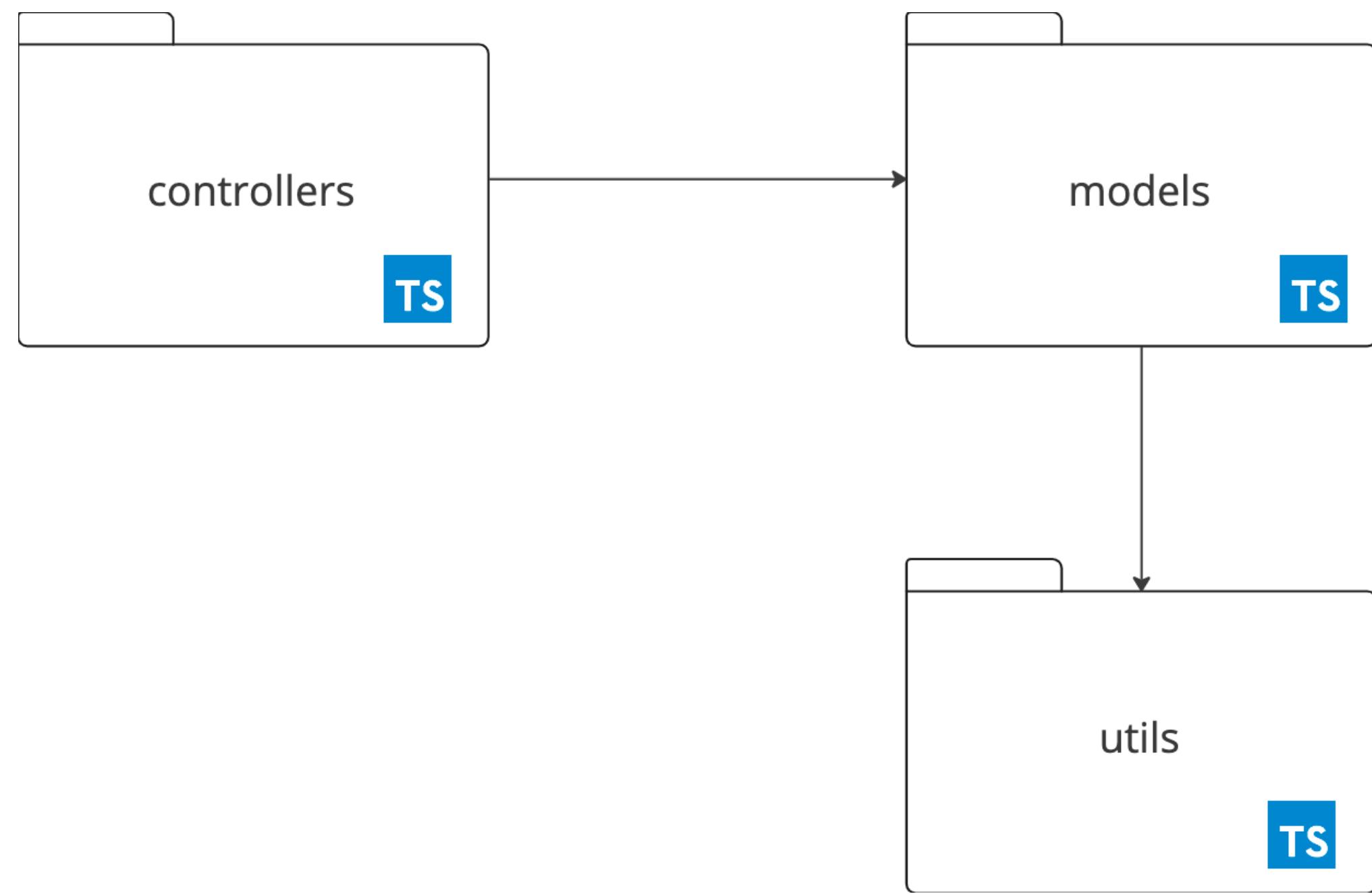
npx tutors-json

```
-bash
Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-json 2.2.3 (tutors-lib: ^0.6.3)
:: Web Development II
  --> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
  --> Introducing Javascript (unit)
:: Introducing Javascript
  --> Module Overview (talk)
  --> JS Introduction (talk)
  --> Variables & Logic (talk)
  --> Const, Let & Objects (talk)
  --> Lab-01 JS Intro(lab)
  --> 1: Introducing Javascript (panelvideo)
  --> 2: Javascript Arrays (topic)
:: 2: Javascript Arrays
  --> Javascript Variables, Objects & Methods (unit)
:: Javascript Variables, Objects & Methods
  --> Variables & Objects Review (talk)
  --> Methods & Arrays Review (talk)
  --> Javascript Arrays (unit)
:: Javascript Arrays
  --> Arrays: Basics (talk)
  --> Array Methods (talk)
  --> Array Iteration (talk)
  --> Arrays of Objects (talk)
  --> Lab-02 JS Arrays(lab)
  --> 02: Javascript Arrays (panelvideo)
  --> 3: Glitch Overview (topic)
:: 3: Glitch Overview
  --> Introducing Glitch (unit)
```

Command generates static site to 'json' folder

```
json
  index.html
  netlify.toml
  <topic-01-js-intro
    panelvideo
    topic.png
  > unit-1
  <topic-02-arrays
    panelvideo
    topic.png
  > unit-0
    talk-e-variables-objects
    talk-f-methods-arrays
    variables-objects-review.png
  <unit-1
    arrays-basics.png
    book-a-js-arrays
    talk-a-arrays-basics
    talk-b-array-methods
    talk-c-array-iteration
    talk-d-arrays-of-objects
  > topic-03-glitch-overview
  > topic-04-web-apps
  > topic-05-app-templates+views
  > topic-06-app-models
  > topic-07-sessions
  > topic-08-apis
  tutors.json
```

“Abstract Syntax Tree”
for course folder/file
structure



```

export class JsonEmitter {
  version = "0.0";

  > emitLo(lo: LearningObject, url: string, jsonObj: any) { ... }

  > emitTalk(lo: Talk, url: string, jsonObj: any) { ... }

  > emitLab(lo: Lab, url: string, jsonObj: any) { ... }

  > emitNote(lo: Note, url: string, jsonObj: any) { ... }

  > emitPanelNote(lo: PanelNote, url: string, jsonObj: any) { ... }

  > emitPanelTalk(lo: PanelTalk, url: string, jsonObj: any) { ... }

  > emitArchive(lo: Archive, url: string, jsonObj: any) { ... }

  > emitPanelVideo(lo: PanelVideo, url: string, jsonObj: any) { ... }

  > emitUnit(lo: Unit, url: string, jsonObj: any) { ... }

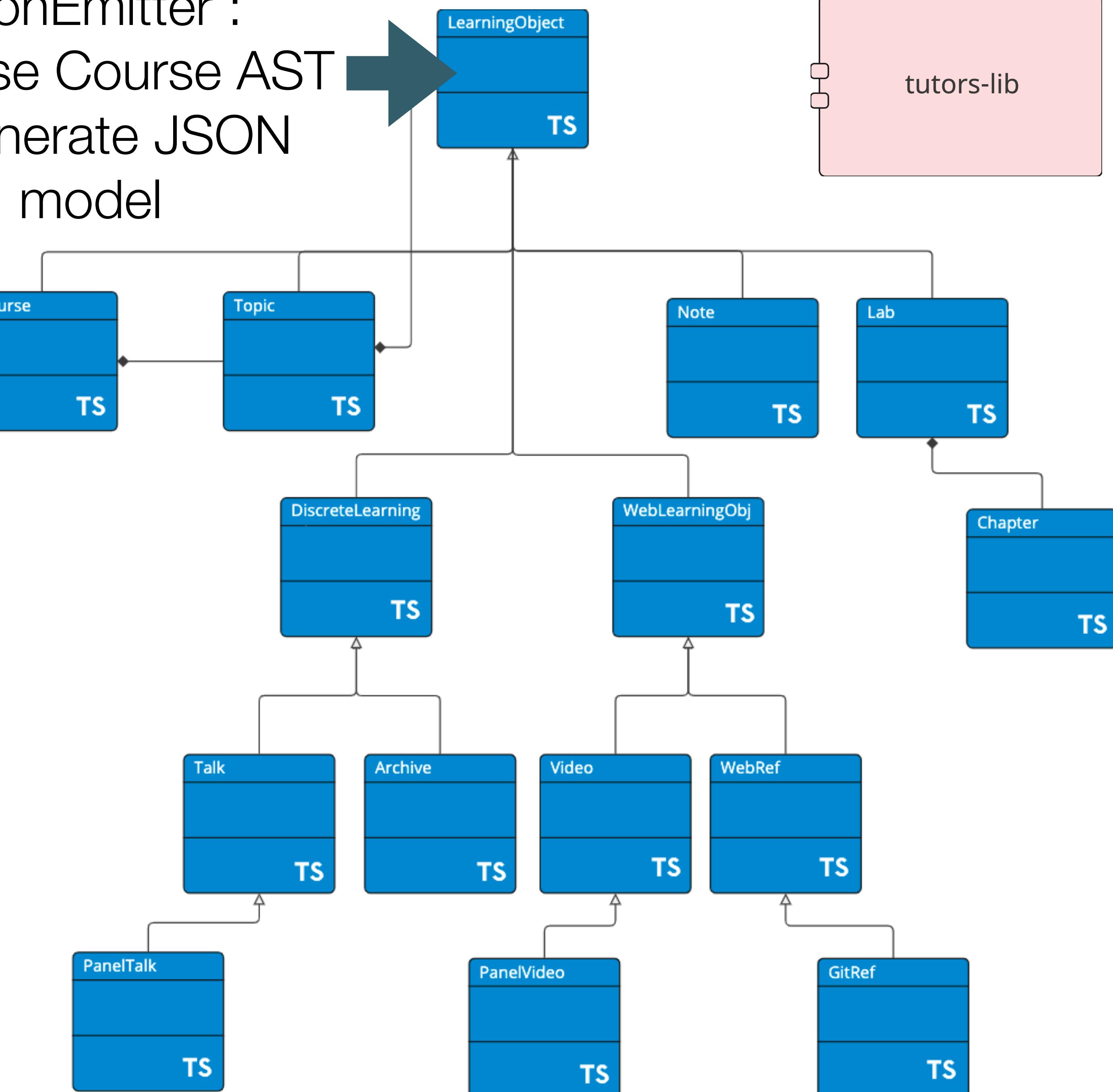
  > emitTopic(lo: Topic, url: string, jsonObj: any) { ... }

  > emitCourse(lo: Course, url: string, jsonObj: any) { ... }

  generateCourse(version: string, path: string, course: Course) {
    const courseJson: any = {};
    courseJson.version = version.toString();
    this.emitCourse(course, "{{COURSEURL}}/", courseJson);
    writeFile(path, "tutors.json", JSON.stringify(courseJson));
  }
}

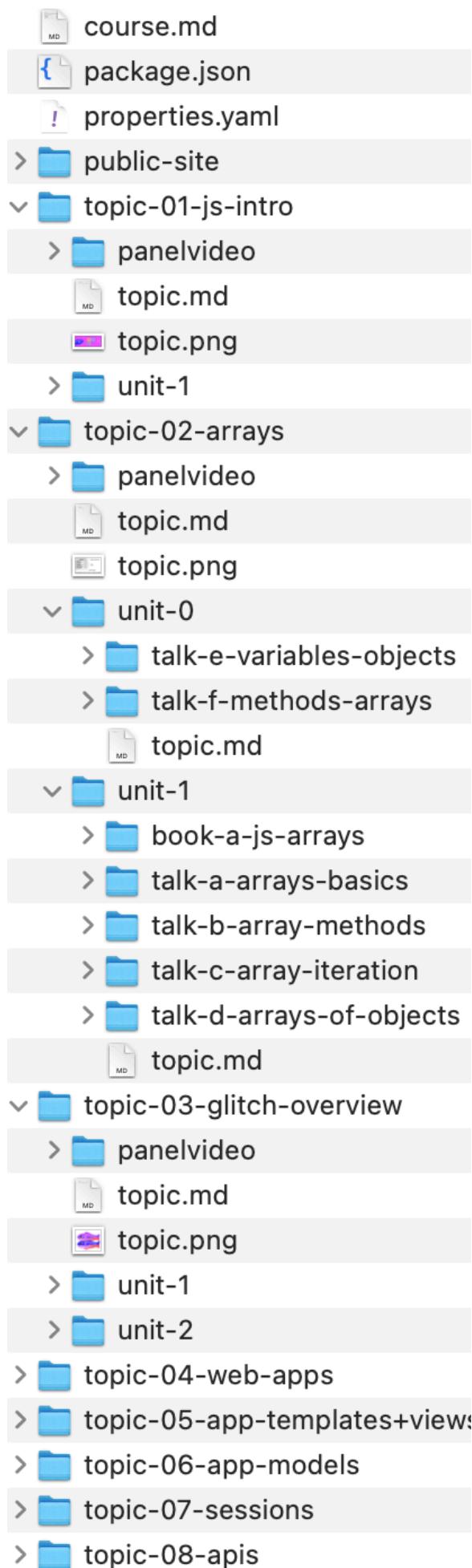
```

JsonEmitter :
Traverse Course AST
+ generate JSON model





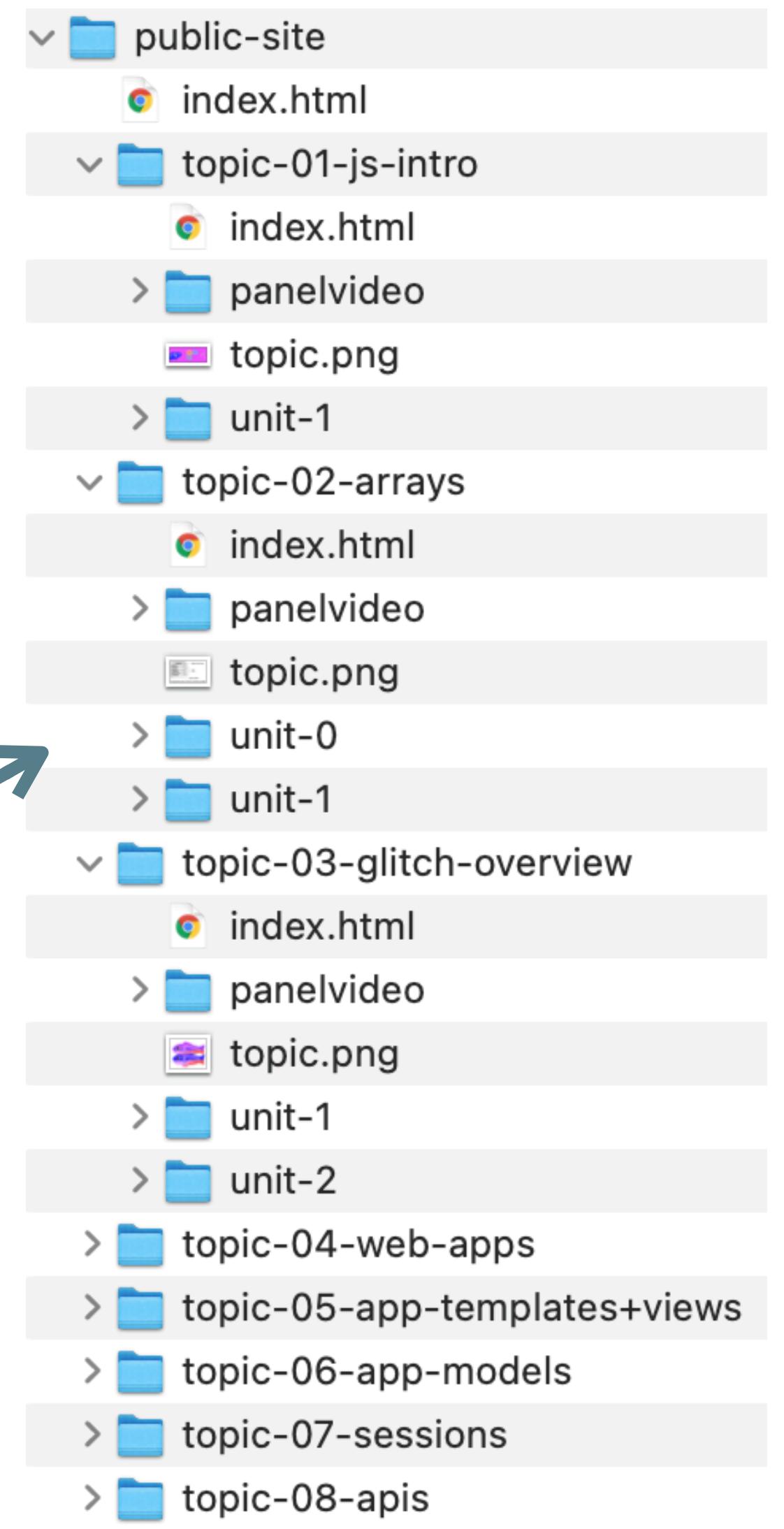
npx tutors-html



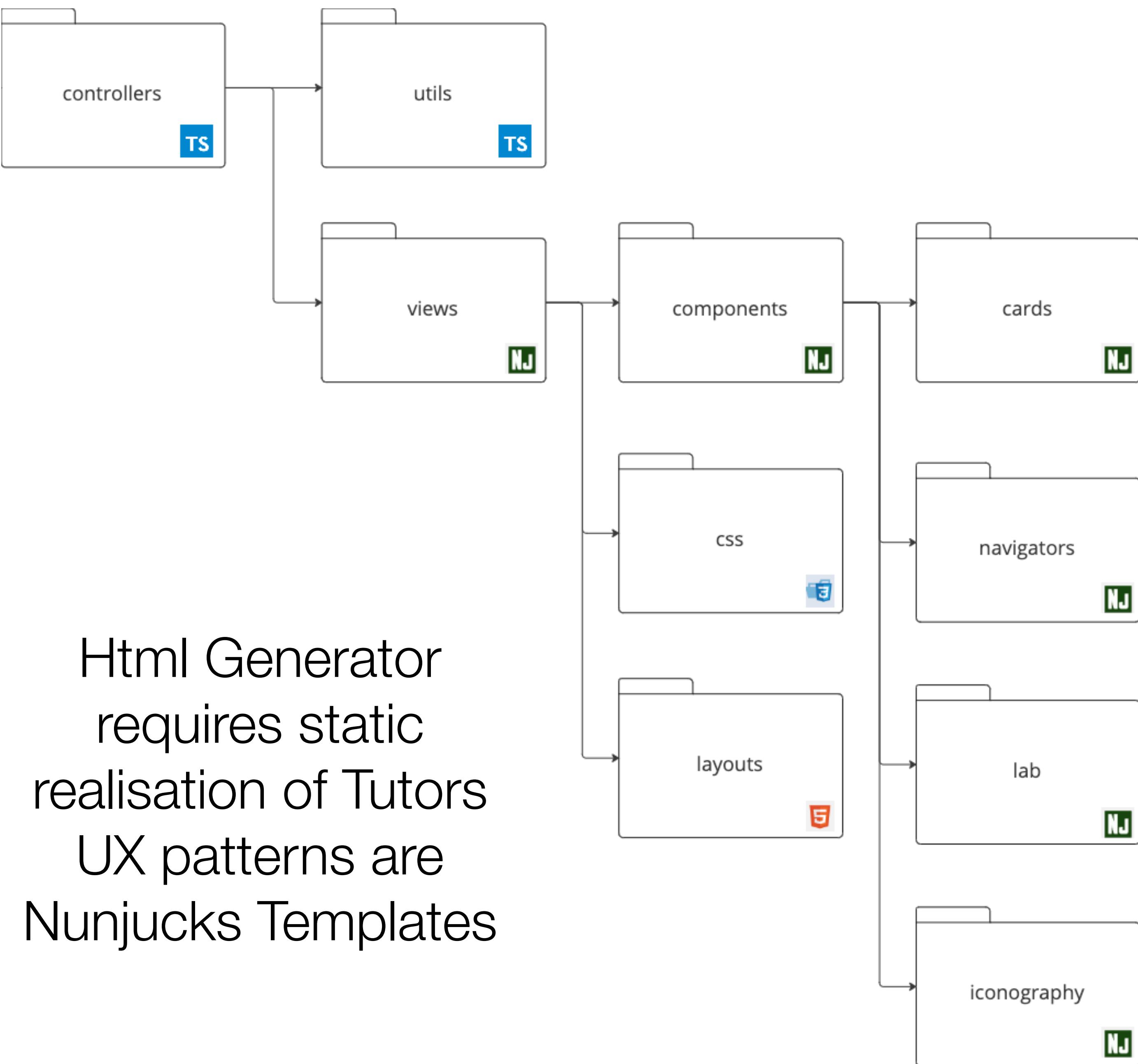
```

Eamonns-Mac-mini:web-development-2-2022 edeleastar$ npx tutors-html
Static course generator tutors-json 0.6.7 (tutors-lib: ^0.6.5)
:: Web Development II
--> 1: Introducing Javascript (topic)
:: 1: Introducing Javascript
--> Introducing Javascript (unit)
:: Introducing Javascript
--> Module Overview (talk)
--> JS Introduction (talk)
--> Variables & Logic (talk)
--> Const, Let & Objects (talk)
--> Lab-01 JS Intro(lab)
--> 1: Introducing Javascript (panelvideo)
--> 2: Javascript Arrays (topic)
:: 2: Javascript Arrays
--> Javascript Variables, Objects & Methods (unit)
:: Javascript Variables, Objects & Methods
--> Variables & Objects Review (talk)
--> Methods & Arrays Review (talk)
--> Javascript Arrays (unit)
:: Javascript Arrays
--> Arrays: Basics (talk)
--> Array Methods (talk)
--> Array Iteration (talk)
--> Arrays of Objects (talk)
--> Lab-02 JS Arrays(lab)
--> 02: Javascript Arrays (panelvideo)
--> 3: Glitch Overview (topic)
:: 3: Glitch Overview

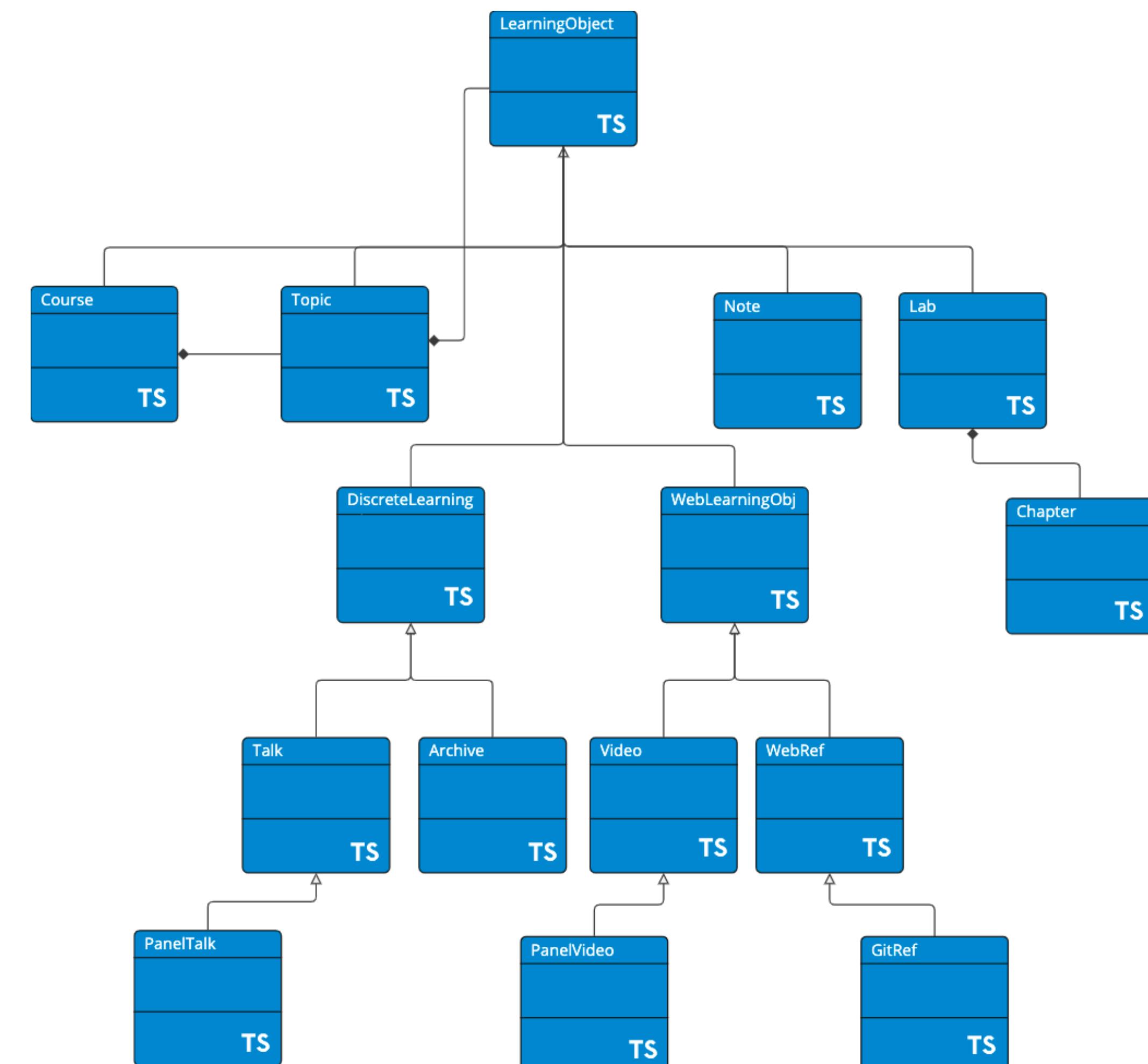
```



Command generates static site to 'public-site' folder



Html Generator
requires static
realisation of Tutors
UX patterns are
Nunjucks Templates



```

export class HtmEmitter {
  parser = new MarkdownParser();

  > emitObjectves(lo: LearningObject) { ... }

  > emitNote(note: Note, path: string) { ... }

  > emitLab/lab: Lab, path: string) { ... }

  > emitUnit(unit: Unit, path: string) { ... }

  > emitLo(lo: LearningObject, path: string) { ... }

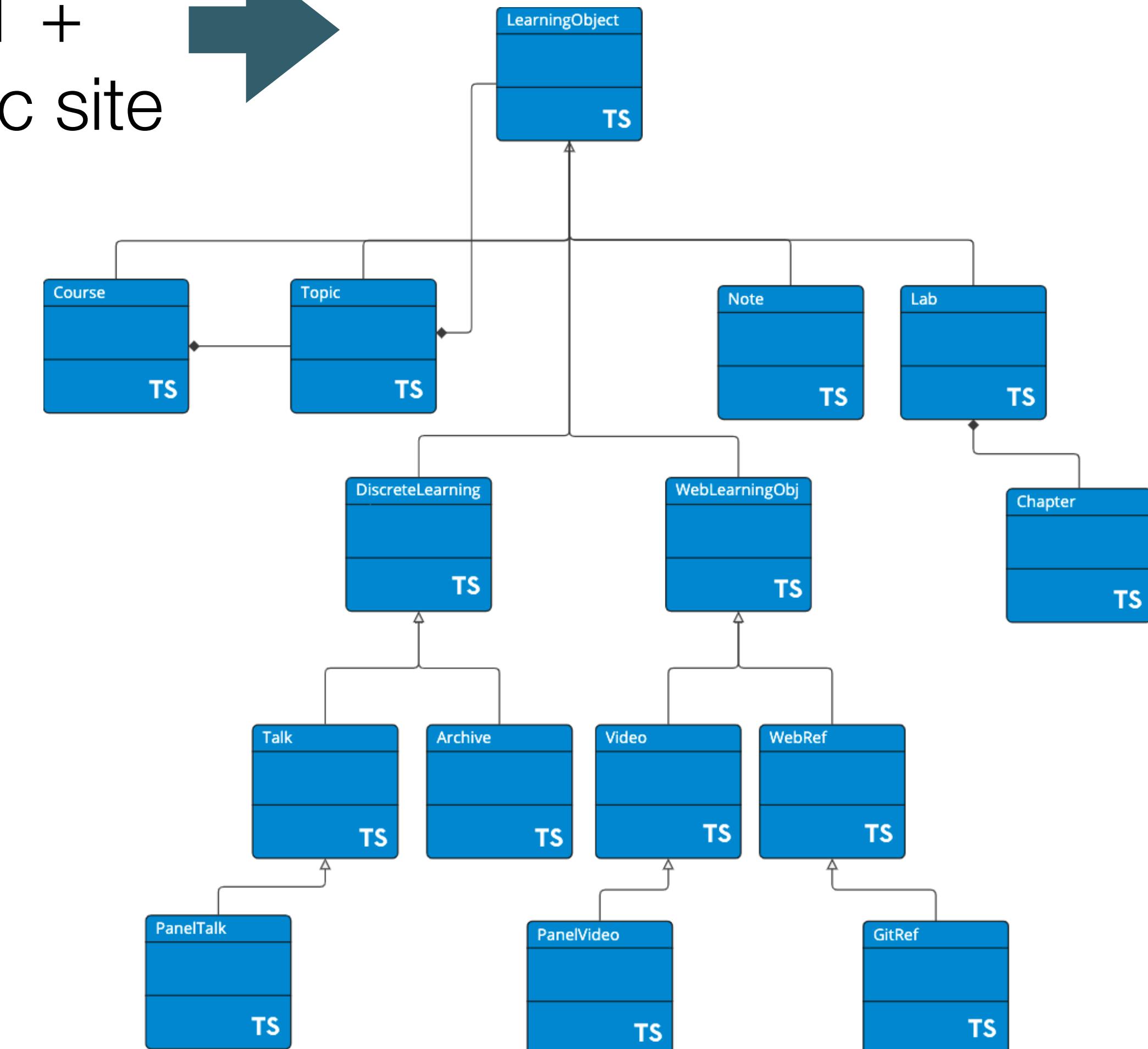
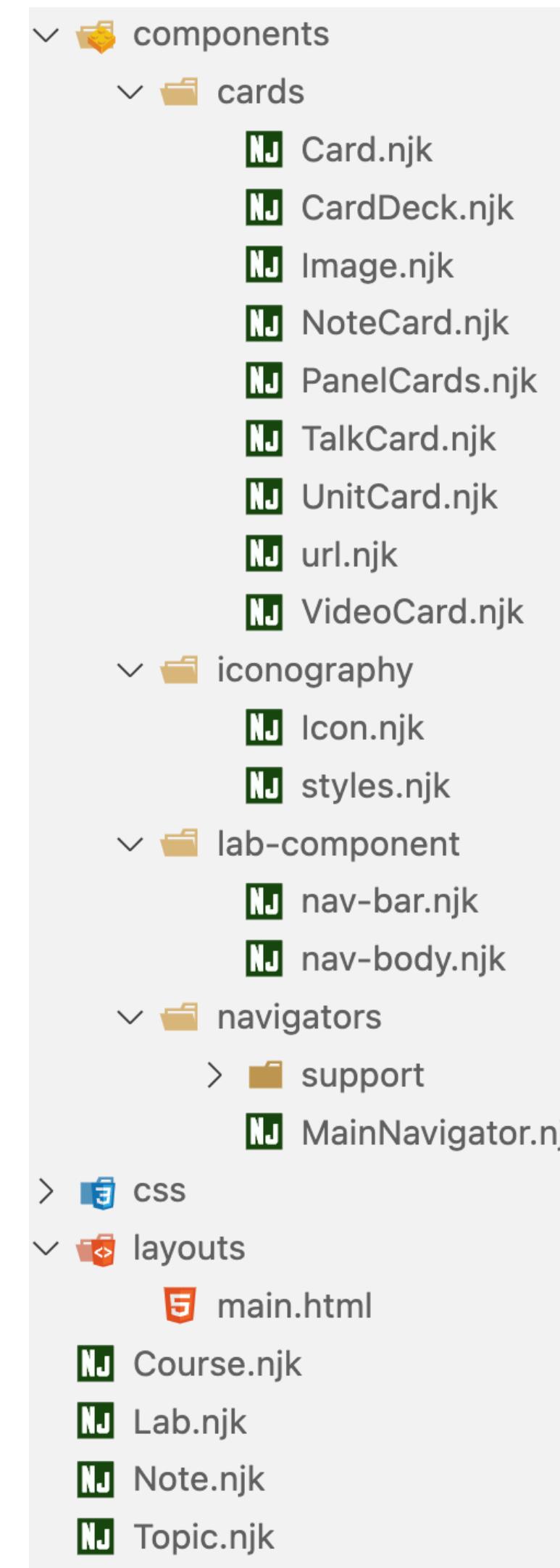
  > emitTopic(topic: Topic, path: string) { ... }

  > emitCourse(course: Course, path: string) { ... }

  generateCourse(path: string, course: Course) {
    sh.cd(path);
    this.emitCourse(course, path);
  }
}

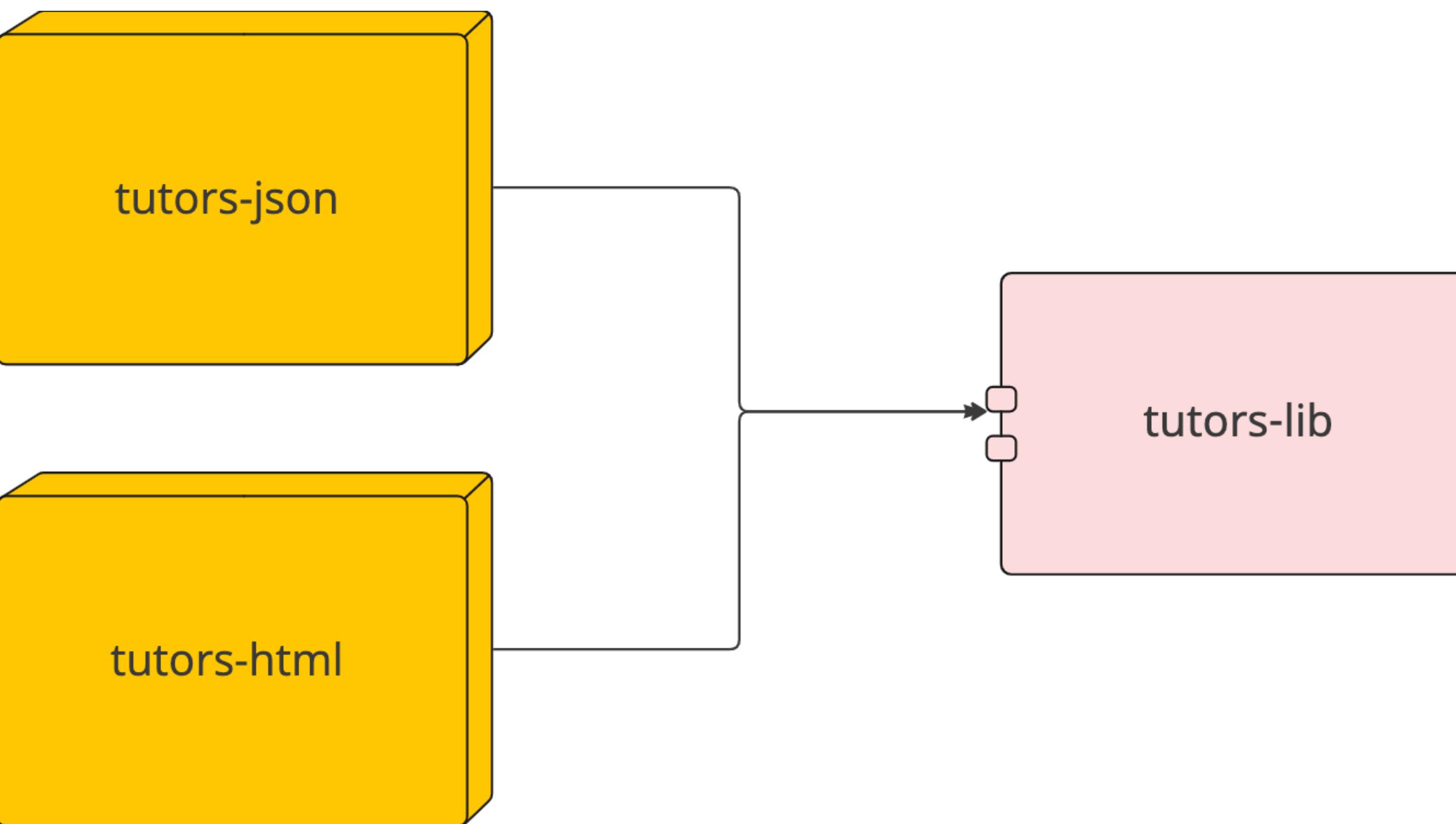
```

HtmEmitter : Traverse Course AST + generate Static site



```
✓ tutros-json
  ✓ src
    ✓ controllers
      TS json-emitter.ts
      TS netlify.ts
      TS tutros-json.ts
    .eslintrc.js
    package.json
    tsconfig.json
```

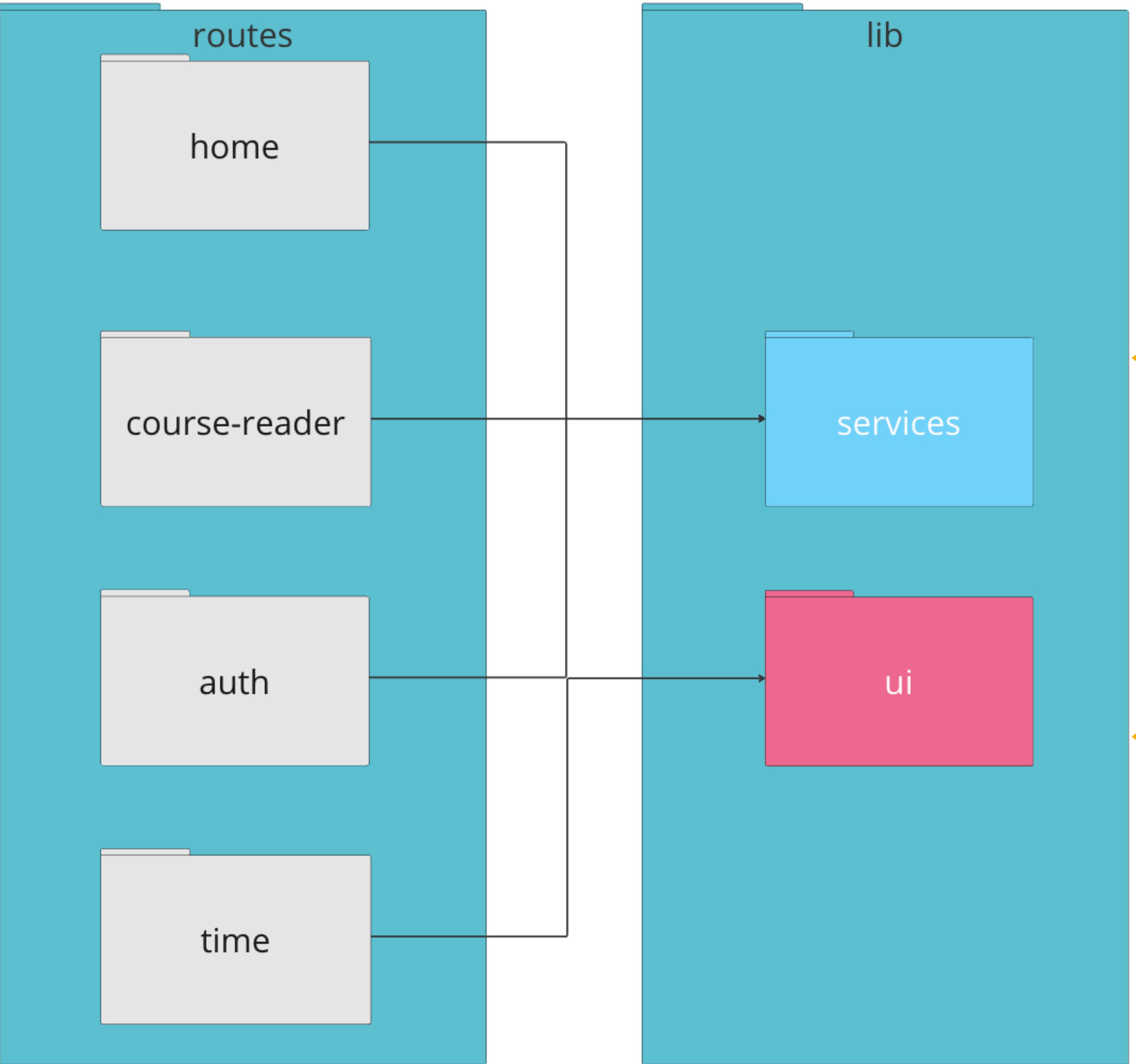
```
✓ tutros-html
  ✓ src
    ✓ controllers
      TS html-emitter.ts
    ✓ utils
      TS markdown-parser.ts
      TS markdown-toc-lib.ts
    ✓ views
      > components
      > css
      > layouts
        NJ Course.njk
        NJ Lab.njk
        NJ Note.njk
        NJ Topic.njk
      TS tutros-html.ts
    .eslintrc.js
    package.json
    tsconfig.json
```



```
✓ tutros-lib
  ✓ src
    ✓ models
      TS course.ts
      TS lab.ts
      TS lo.ts
      TS los.ts
      TS note.ts
      TS topic.ts
      TS web-los.ts
    ✓ utils
      TS futils.ts
      TS lutils.ts
      TS properties.ts
      TS videoutils.ts
    .eslintrc.js
    package.json
    tsconfig.json
```

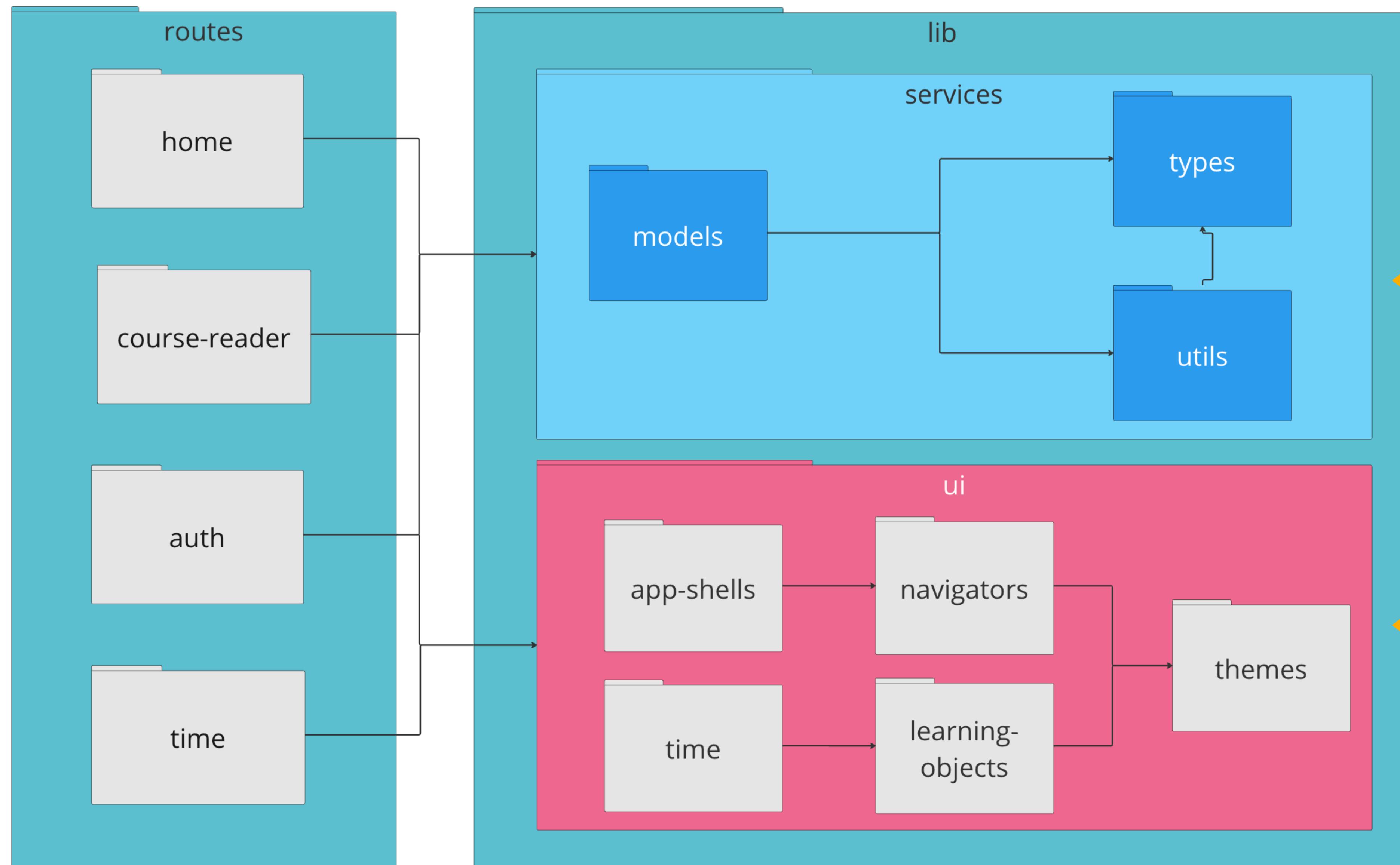
Reader Architecture

SvelteKit Web Application



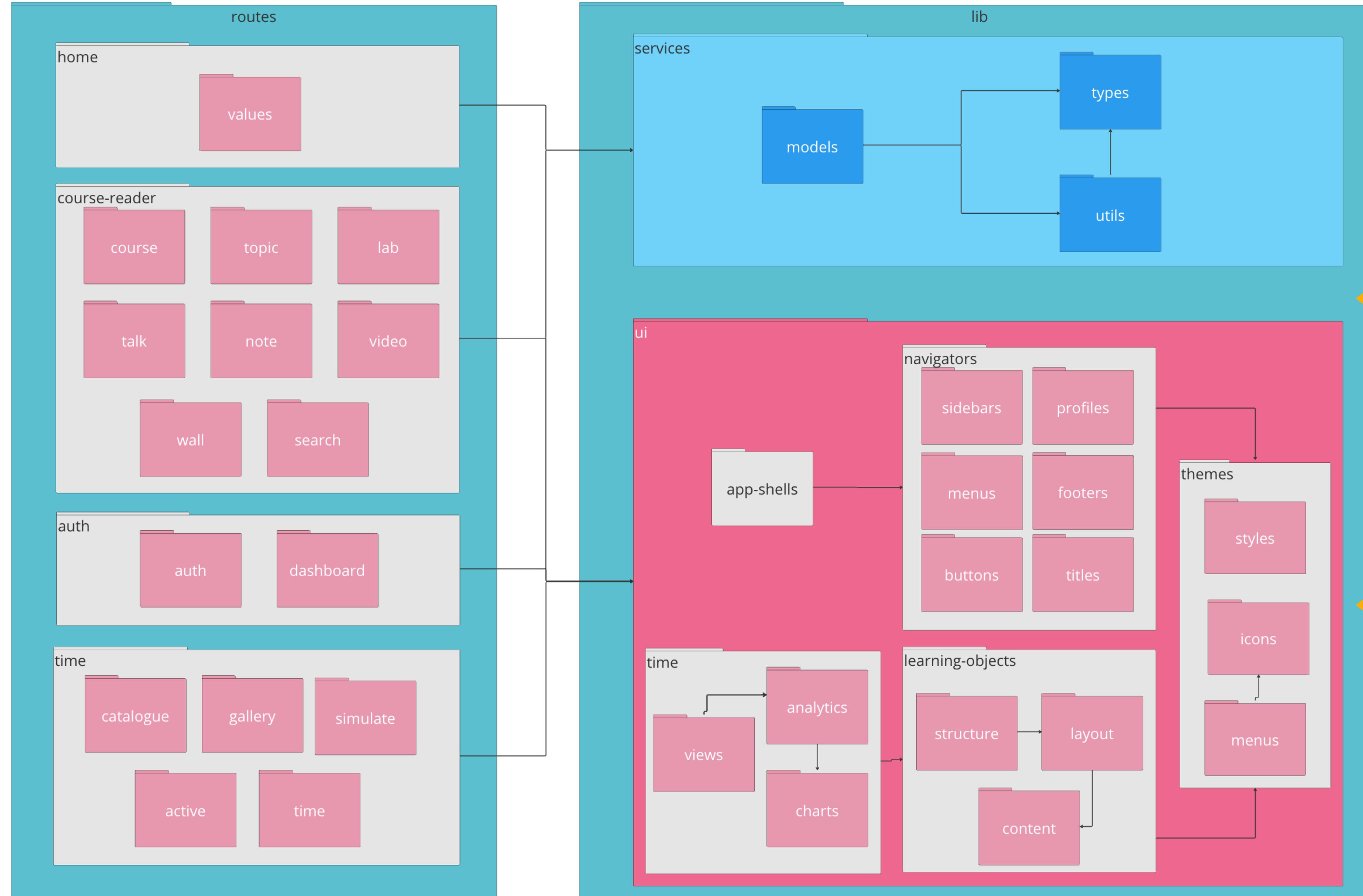
Course Models
Analytics,
Metrics &
Presence
systems

UI
Components



Course Models
Analytics,
Metrics &
Presence
systems

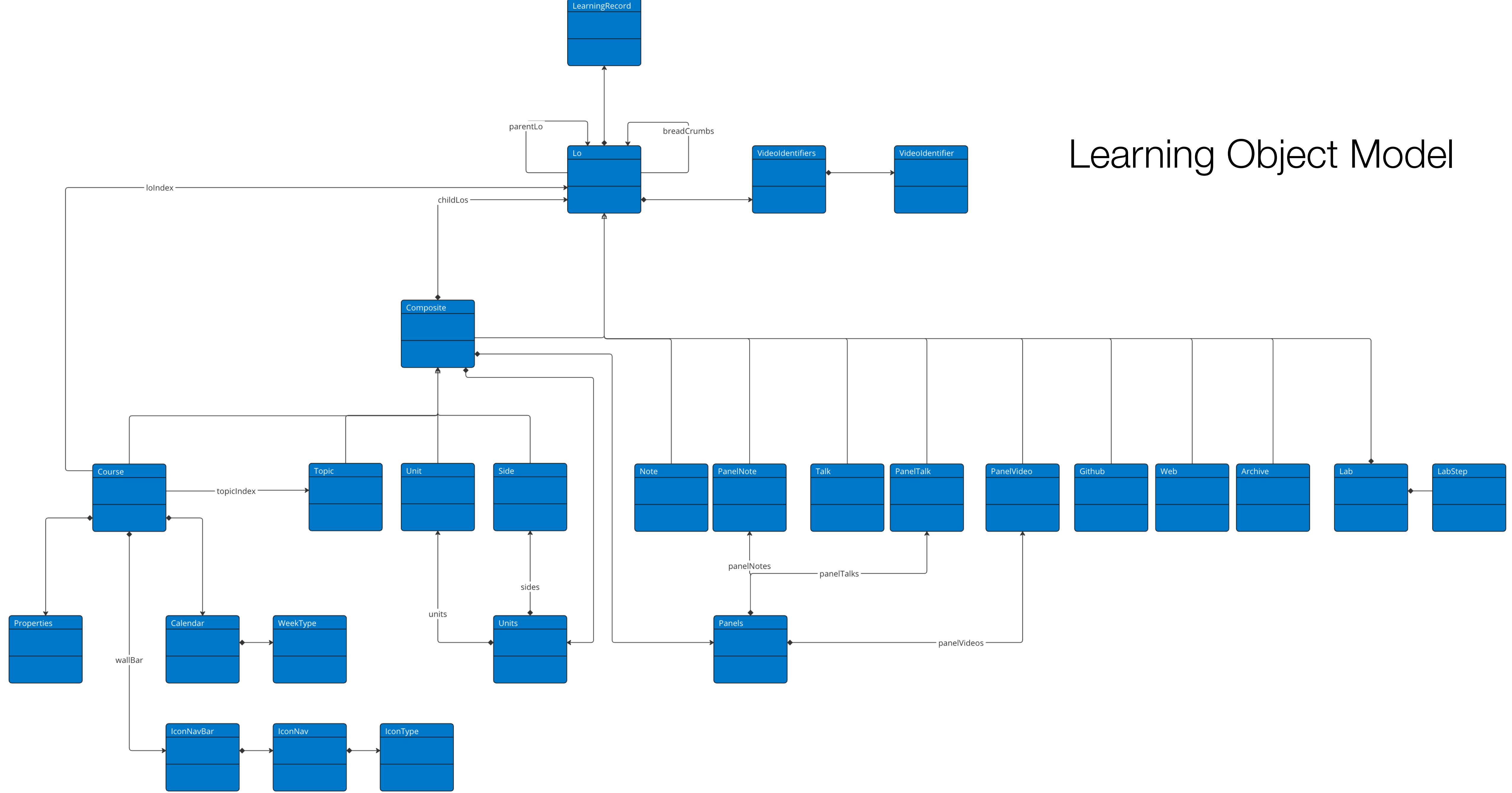
UI
Components



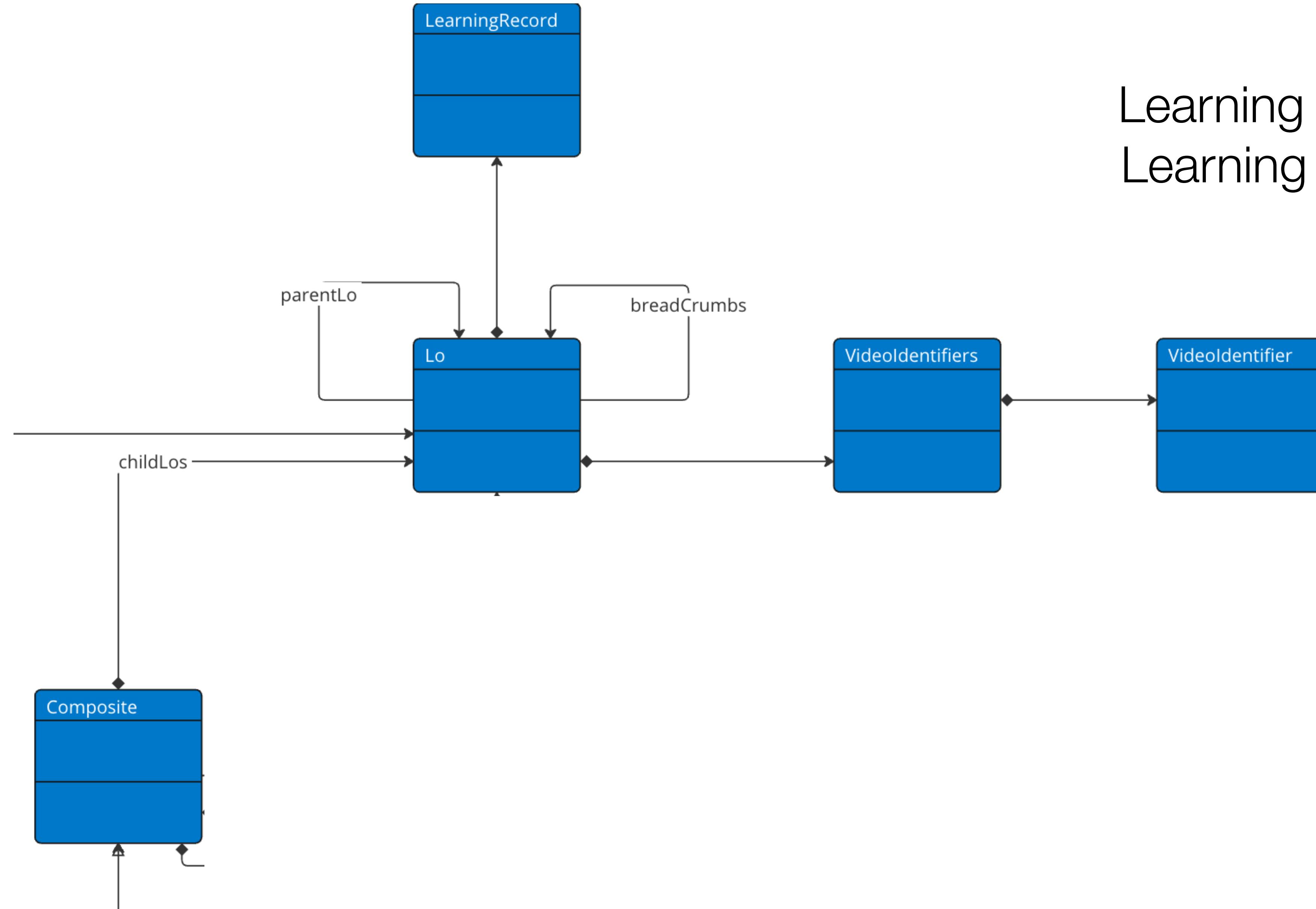
Course Models
Analytics,
Metrics &
Presence
systems

UI
Components

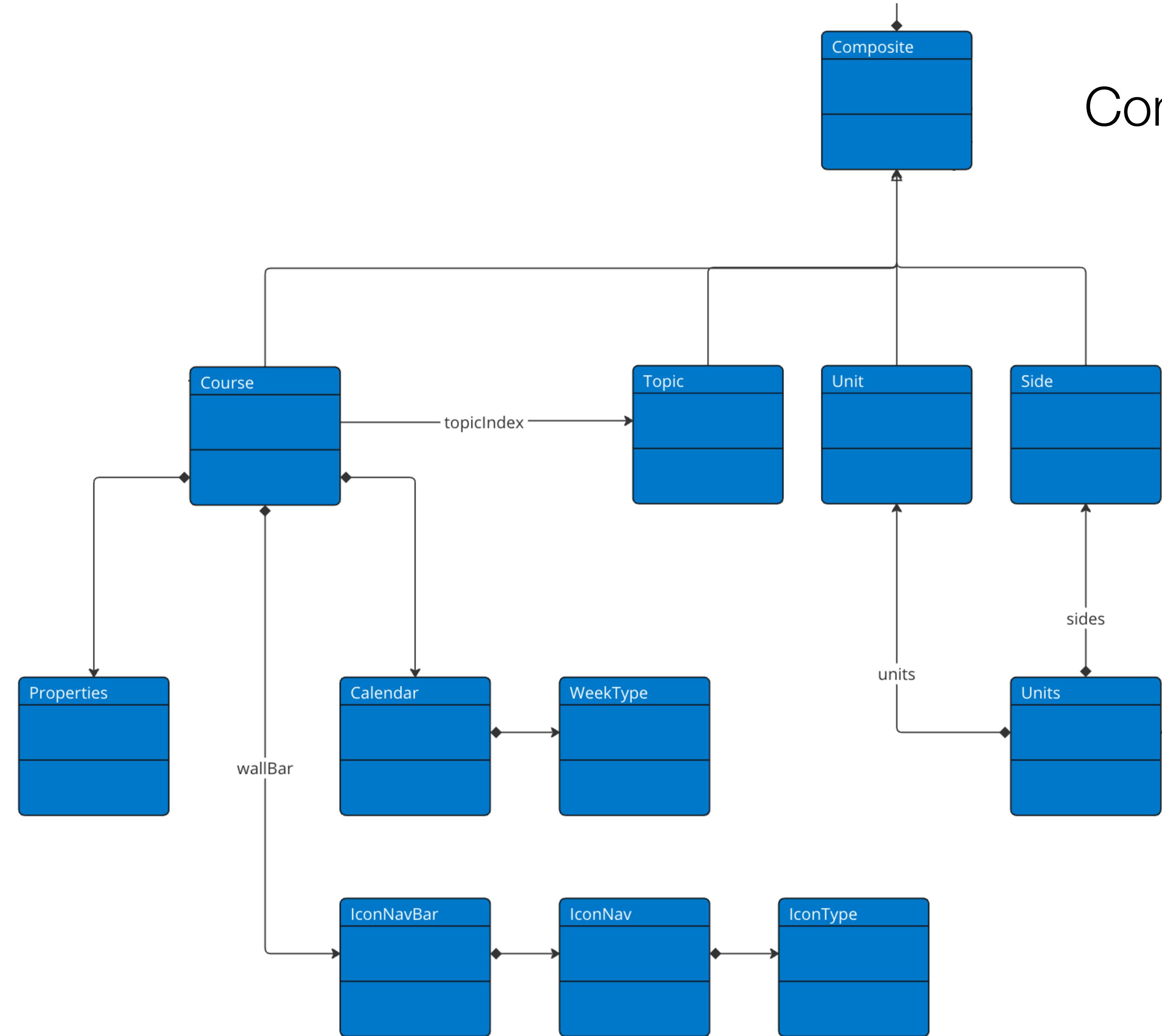
Learning Object Model



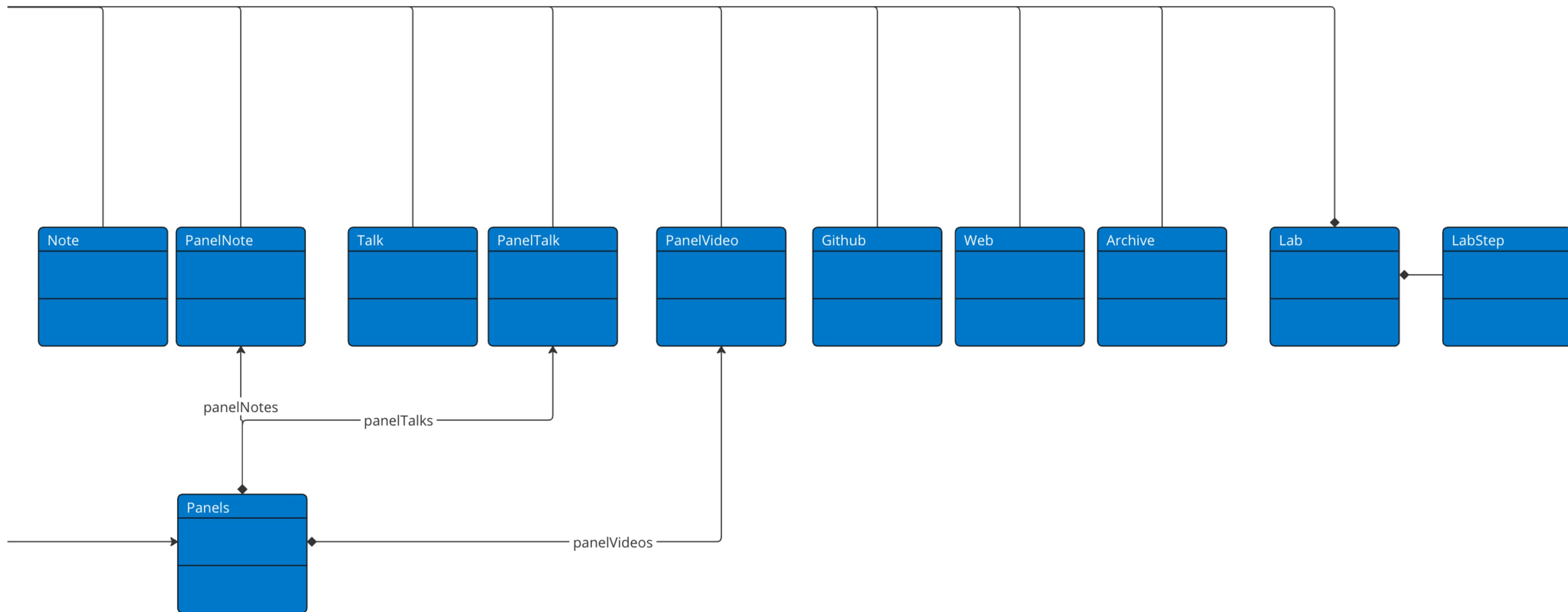
Learning Object & Learning Records



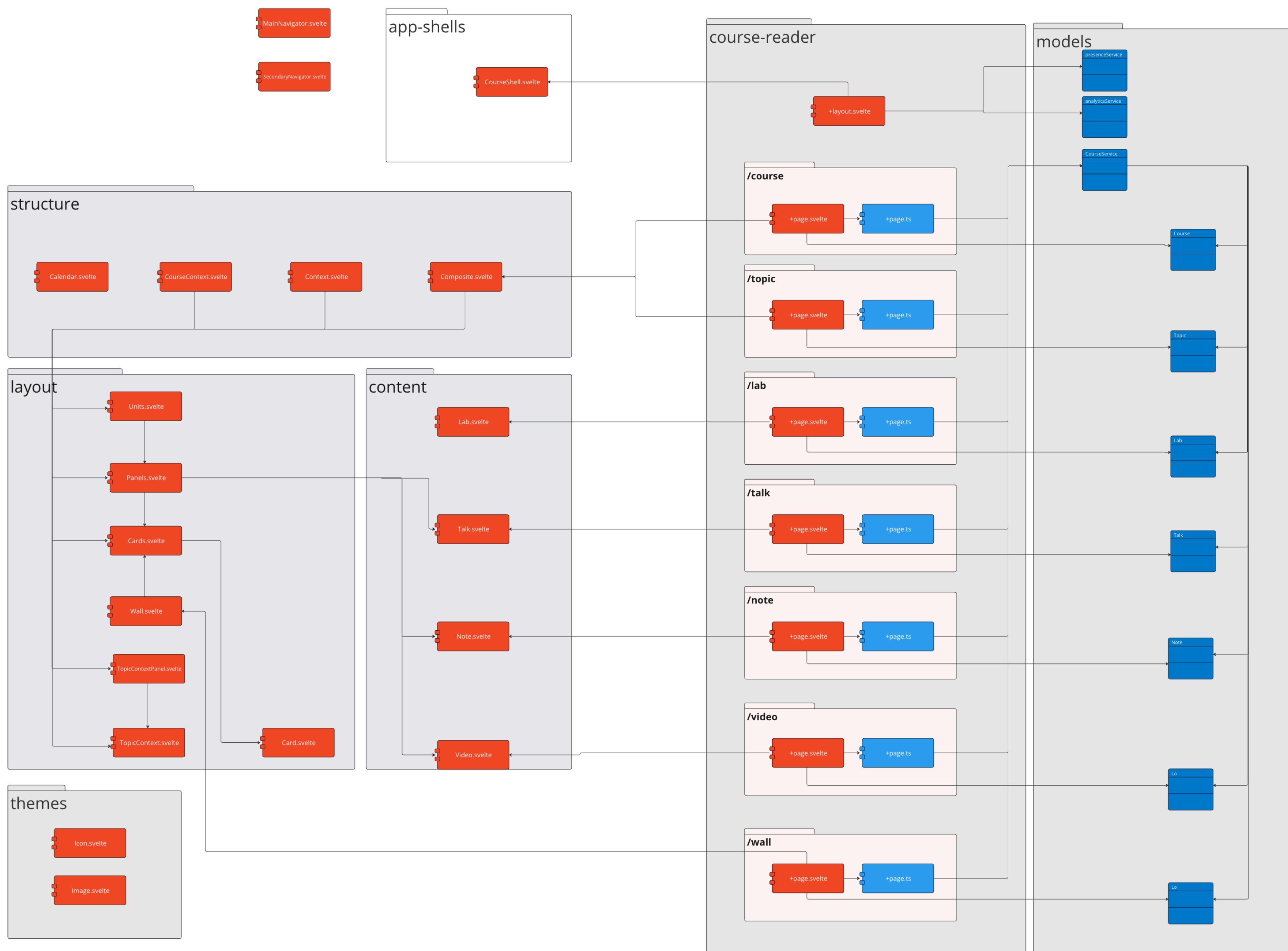
Composite Learning Objects



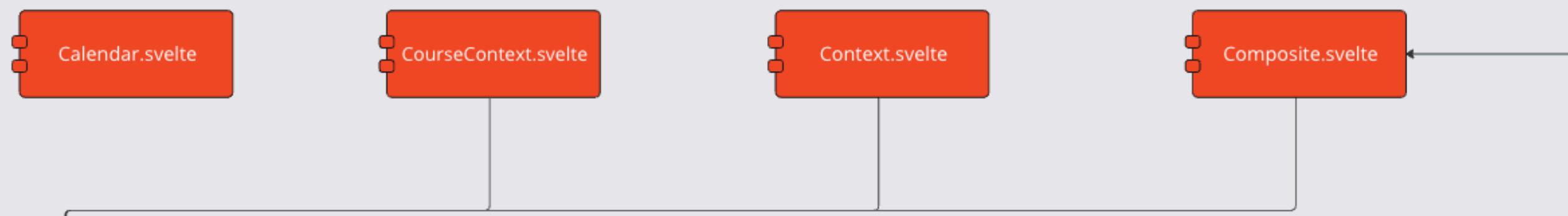
Learning Object Model



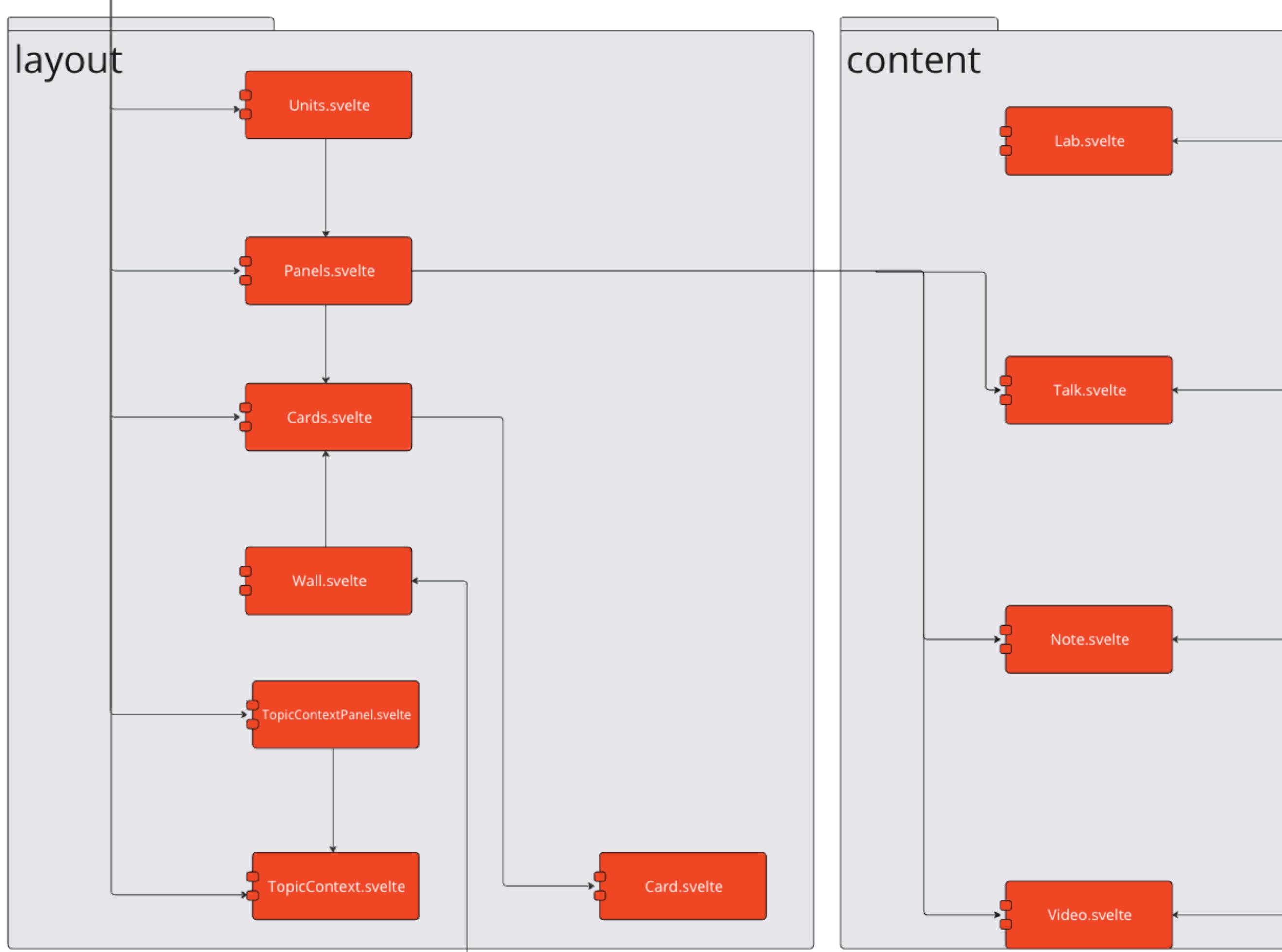
Course Routes



structure

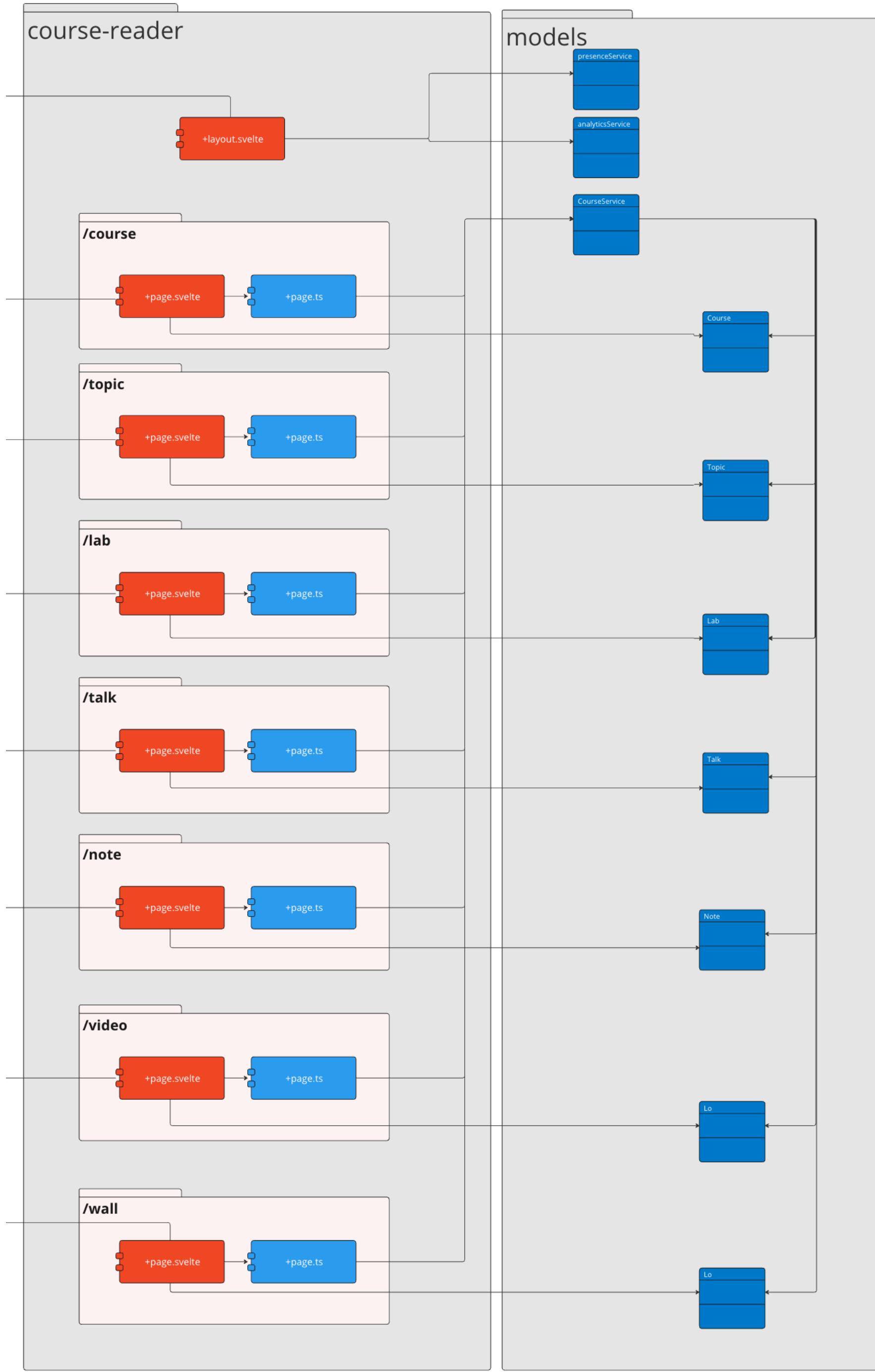


layout



Visual Learning Object Model

Learning Object Routes



Learning Objects

Semester 1: January-June 2021

Workshop One: induction · structure · schedules · handbook

Life etc...: time · space · balance · experiences · tips · techniques

Web Development: html · css · layout · web apps · web frameworks · deployment

Programming: algorithms · data structures · processing · java · classes · libraries

Semester 1 Schedule: timetables · semester calendars · assessment schedules

Semester 2: June-December 2020

Workshop Two: introducing semester Databases & Computer Systems - project &...

ICT Skills 1: Javascript · node · express · git · github · glitch

Computer Systems & Networks: logic · computer organisation · os · networks · interfaces · sensors

Database: entities · tables · rows · sql · er · nosql

Semester 2 Schedule: timetables · semester calendars · assessment schedules

Semester Three

Workshop Three: agile methods · cv preparation · introducing semester 3

Full Stack Web Development: javascript · node · apis · tdd · frameworks · front-end · svelte

Software Security: crypto · vulnerabilities · authentication · pentesting

Developer Operations: cloud computing · scripting · scaling · automation · monitoring

Semester 3 Schedule: timetables · semester calendars · assessment schedules

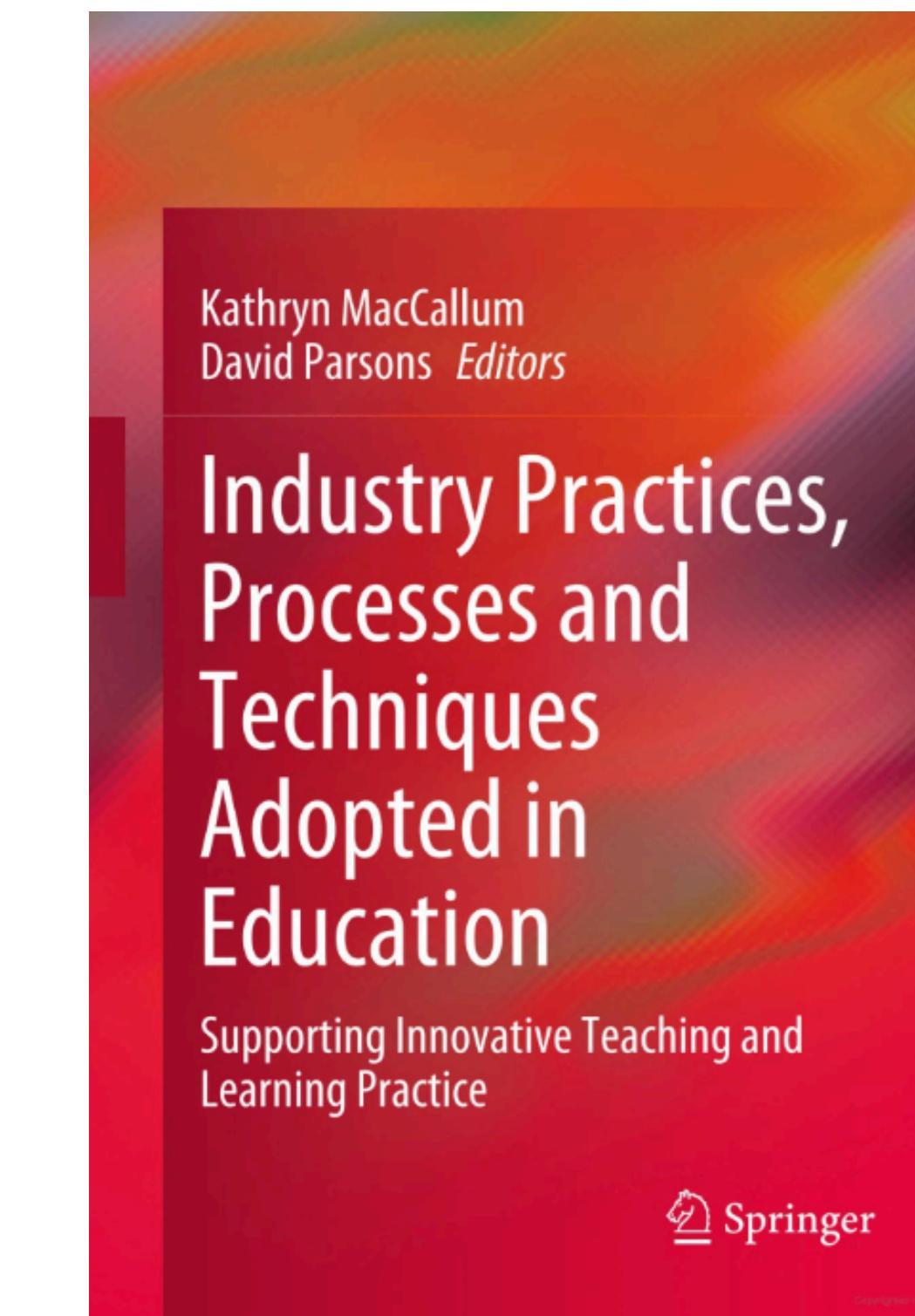
Semester 4: June-December 2021

Workshop Four: Introducing semester 4, Project, Placement

ICT Skills 2: javascript · SPA · react · APIs · front-end

Mobile App Development: layouts · activities · resources · lifecycle · widgets · ux

Semester 4 Schedule: timetables · semester calendars · assessment schedules



The Agile Semester: How we used Agile as an engagement mechanism

Leigh Griffin, Brendan O'Farrell, Colm Dunphy, Peter Windle, Eamonn de Leastar

Leigh Griffin, Brendan O'Farrell

Red Hat, Communications House, Cork Road, Waterford City, Ireland, X91NY33, +35351810111 e-mail: lgriffin, bofarrell @redhat.com

Colm Dunphy, Peter Windle, Eamonn de Leastar

Waterford Institute of Technology, Cork Road, Waterford City, Ireland, X91K0EK, +35351302000 e-mail: cdunphy, pwindle, edeleastar @wit.ie

Abstract

Red Hat is a global leader in open source technology, and Agile transformations in distributed environments. Waterford Institute of Technology (WIT) is a college that launched a world leading ICT conversion course - HDip in Computer Science, delivered exclusively online. Red Hat partnered with WIT, to help shape the evolution of the delivery of the HDip over the last 5 years. This chapter shares industry best practices for remote Agile practice and highlights the pitfalls in remote engagement, and cultural change. The adoption of the Scrum framework and its impact on the people and processes is outlined and key lean tools are highlighted. This case study views Agile adoption and implementation through the lens of several years of delivery of an online programme, exploring the drivers for engagement from personnel, instructors, and students; prioritising of individuals and interactions over processes and tools; how value is delivered incrementally, and the paradigm shift required from established education system norms in delivering a new approach - The Agile Semester. The creation and use of tooling to support this approach, and its alignment with educational theories is discussed; TutorStack framework, and Tutors open source project. Three Insight adaptors for TutorsTime are examined highlighting their benefits over current practice – Lab Progress; Academic Calendar Activity; and Semester Profile. Key examples of the importance of community building, the student experience, and agile delivery & planning in education are explored. An educational lens is applied to The Agile Manifesto. Conclusions and future work are outlined.

