

Building Modern Web Applications & Services using Node.js



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



Eamonn de Leastar
edeleastar@wit.ie

<https://wit-oth-regensburg-2017-dmas.github.io>

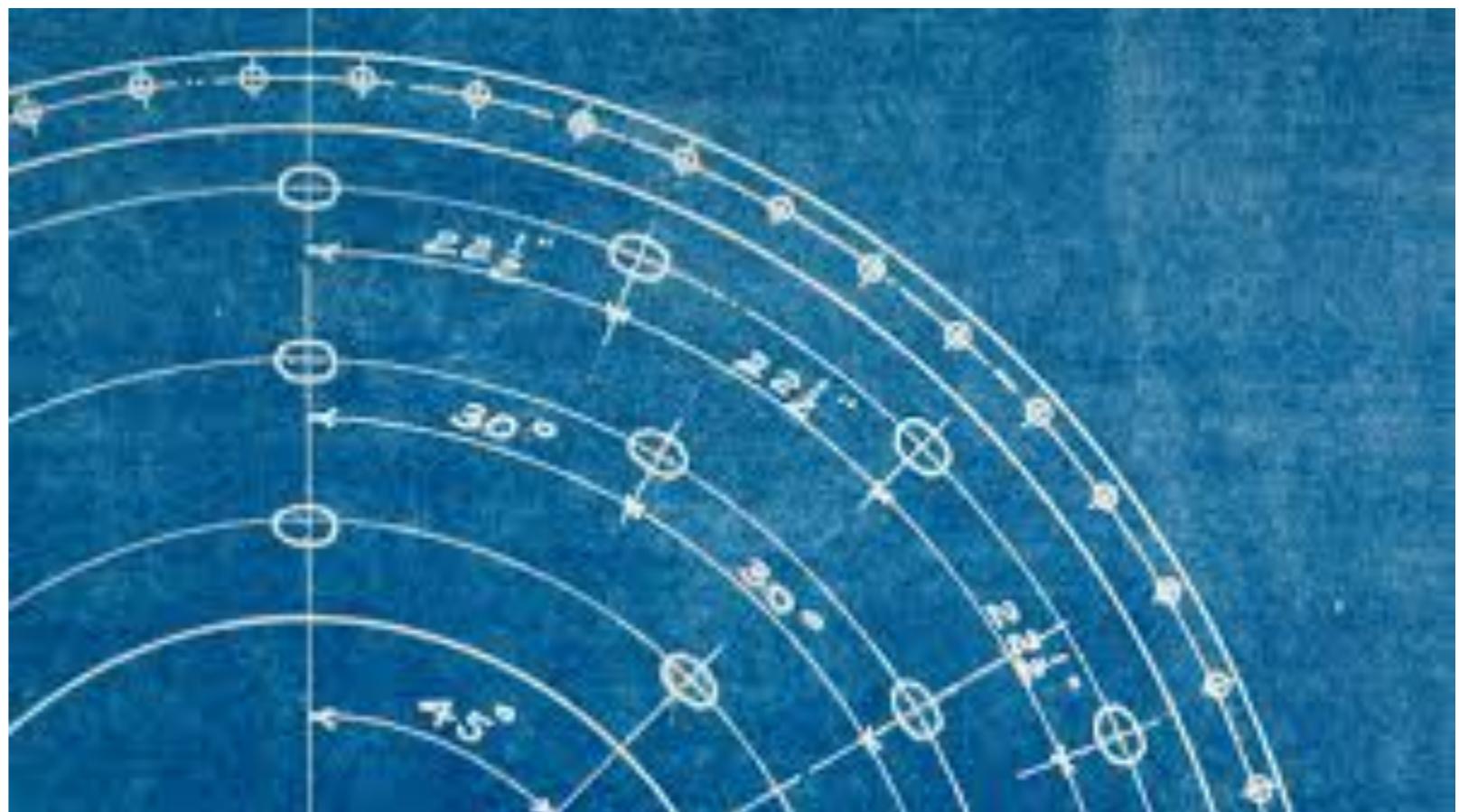
Course Mission

Transfer a set of foundation skills to enable you to design, build, secure, test and deploy a modern web application + API.



Agenda

- Prerequisites
- Preparing for the course
- Brief Overview
- Lab Requirements
- Assessment Guidelines
- Schedule



Perquisites

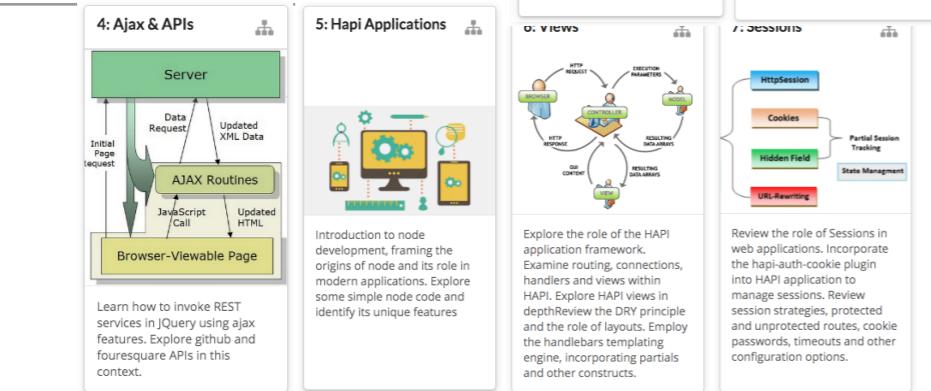
- Foundation level skills in:
 - HTML: Ability to effectively structure the html content of a small to medium static site, including the use of templates
 - CSS: Understand the fundamentals of CSS, and be able to realise simple layouts and designs
 - Javascript: Be familiar with the building blocks of the language and be able to compose realise algorithms to accomplish simple tasks.



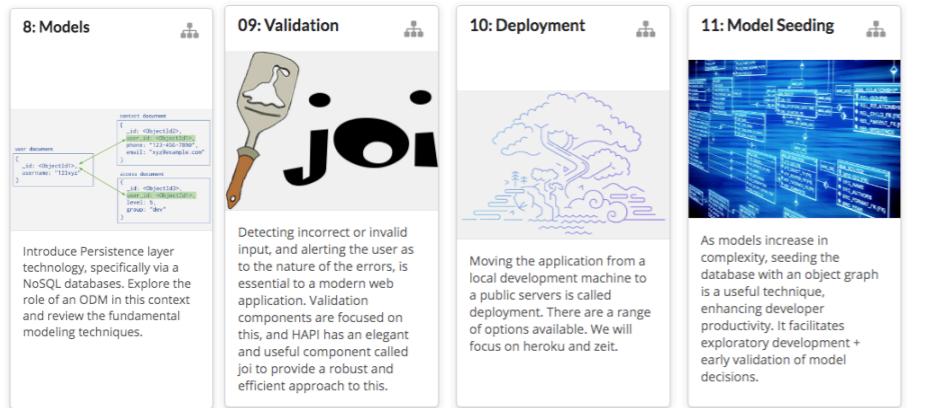
5 Topics Top Level Topics



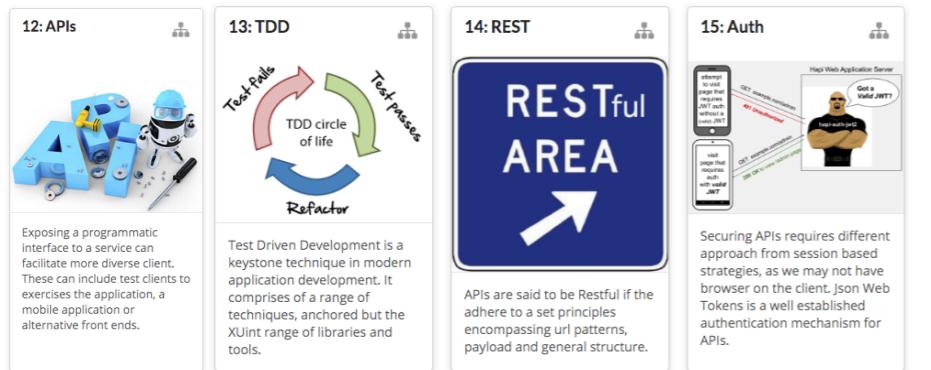
(1) Front End Foundation (+JS)



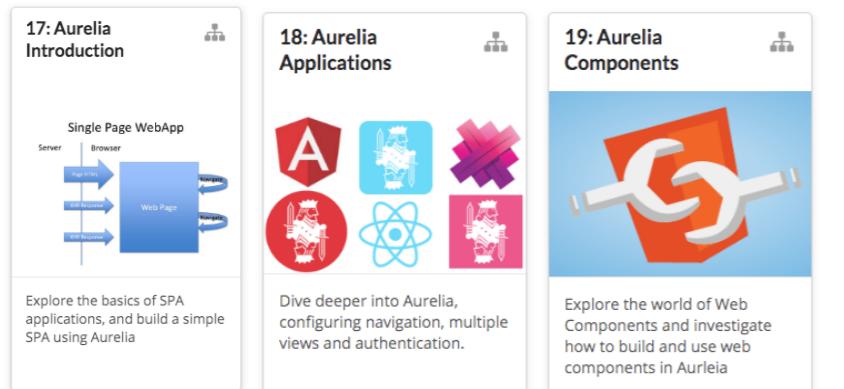
(2) Apis, Node & Hapi Applications



(3) Models, Persistence & Deployment



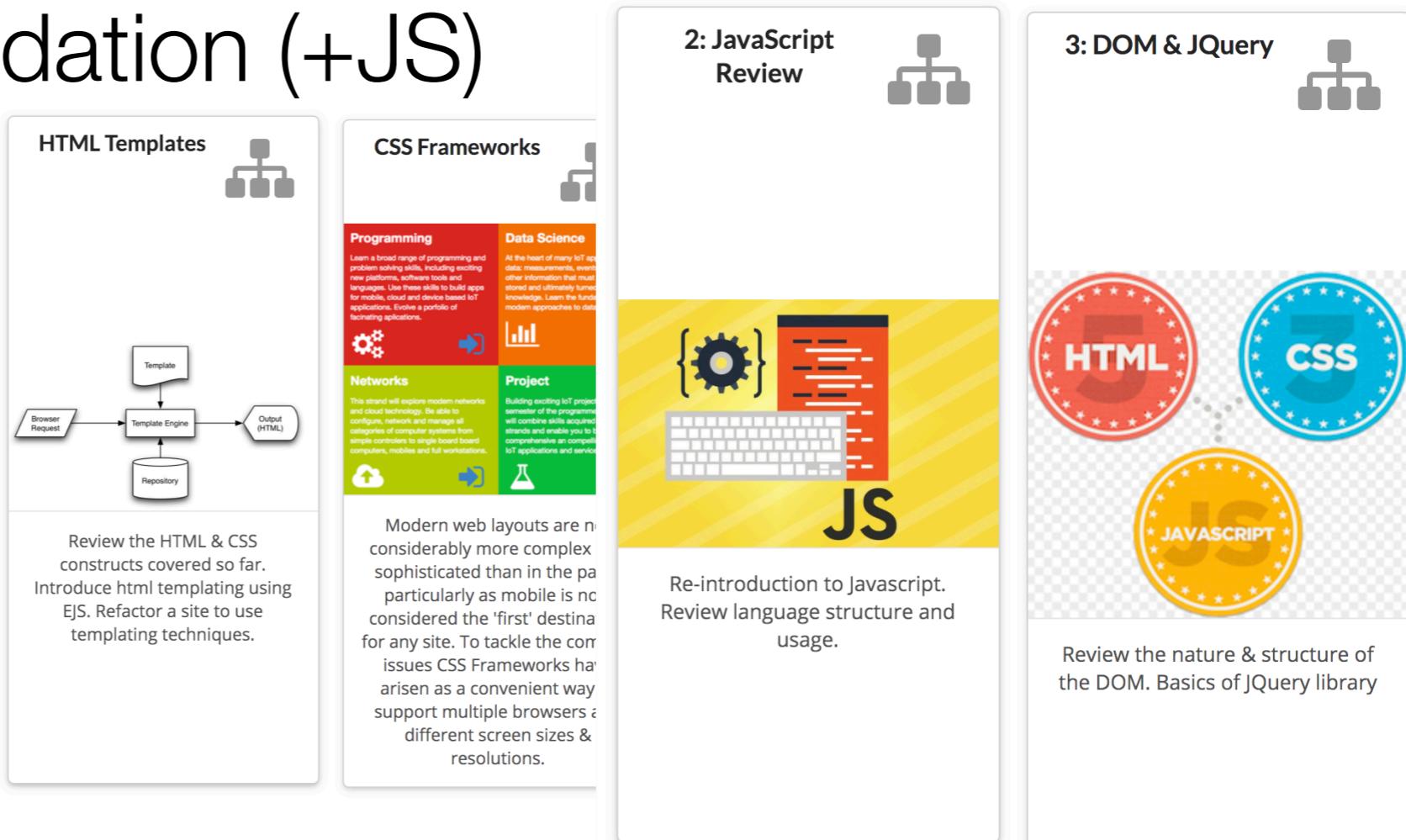
(4) Test Driven API Development



(5) Single Page Applications (TypeScript)

(1) Front End Foundation (+JS)

- Be able to structure and style a simple web site using html5, templating + a CSS framework.
- Understand the fundamentals of Javascript & JQuery



Concepts

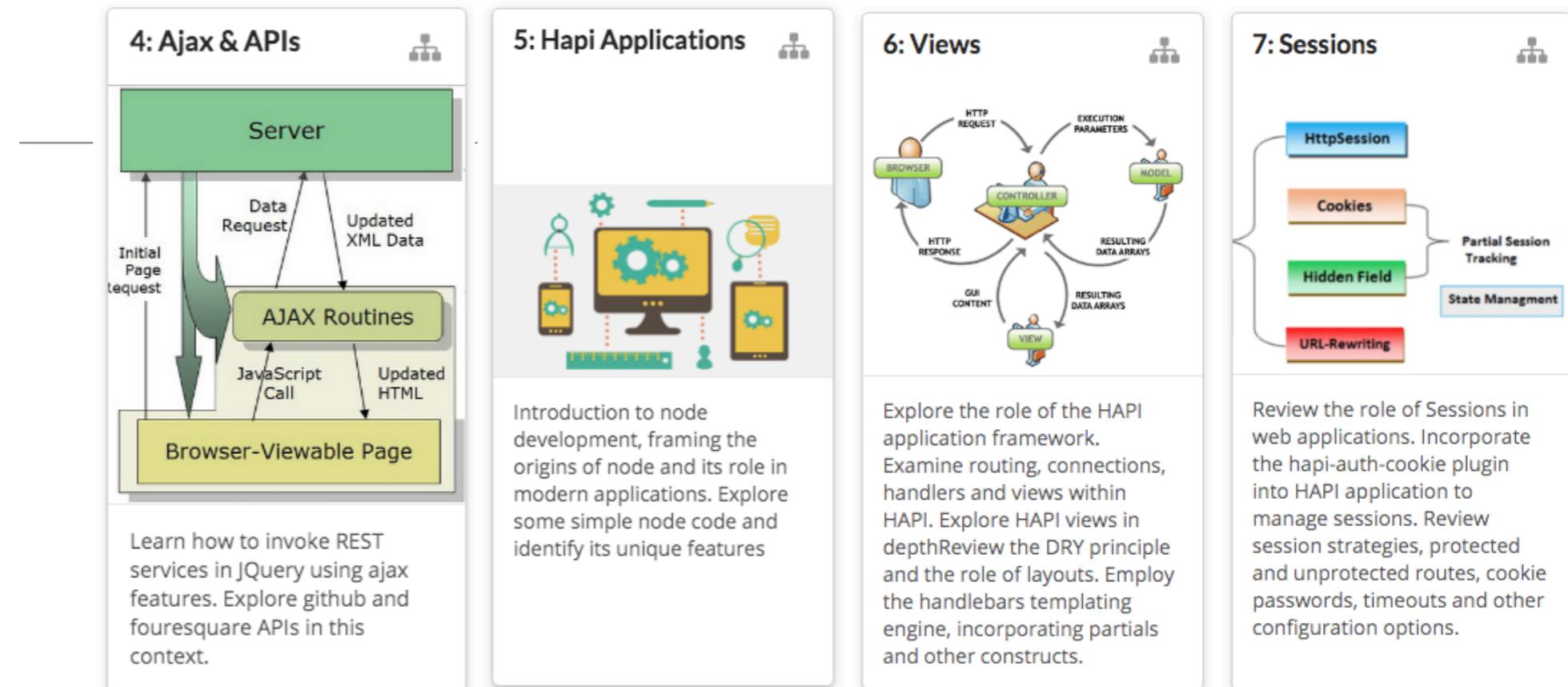
- Review Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery

Tools:

- Html5
- Semantic-ui
- jQuery
- Chrome dev tools
- DOM
- WebStorm IDE

(2) Apis, Node & Hapi Applications

- Be able to build a simple Node application incorporating templates views
- Understand and use the fundamentals of session management in the application



management in the application

Concepts

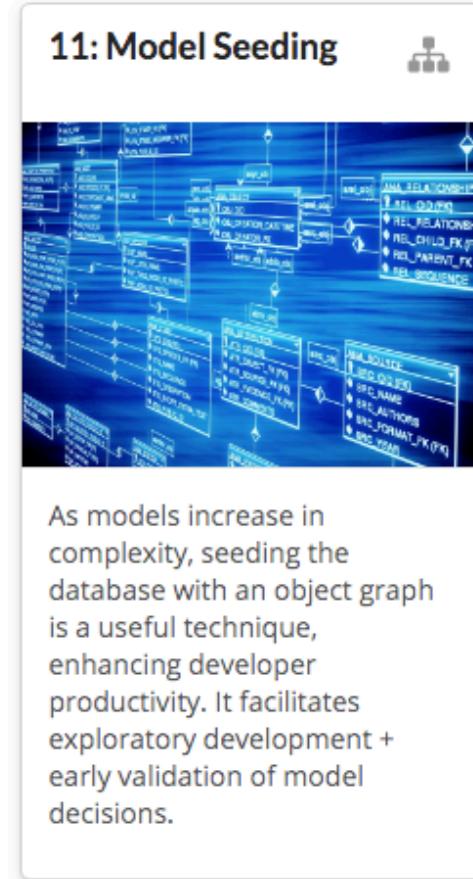
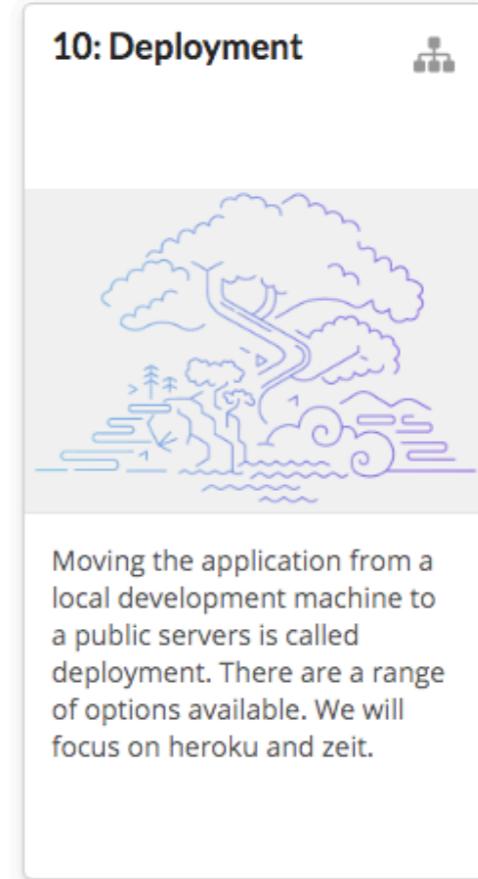
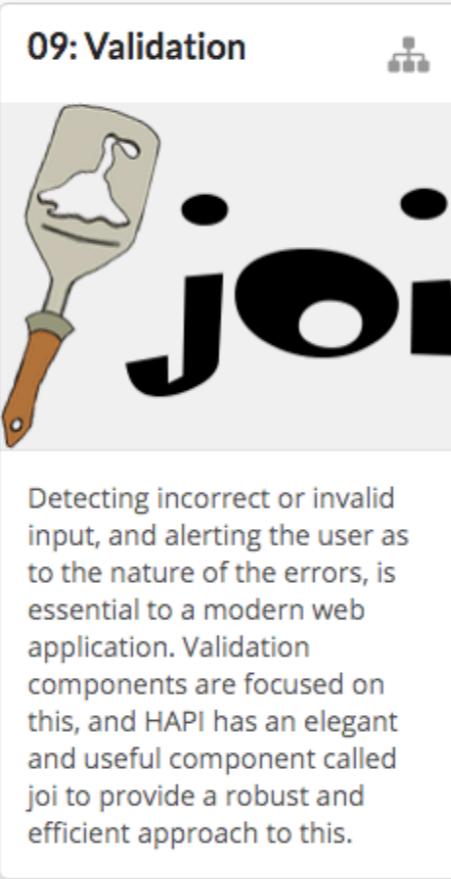
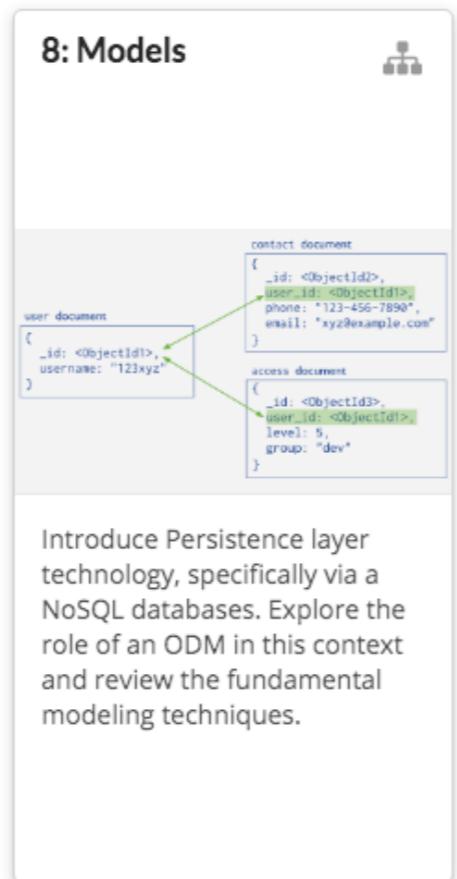
- Accessing APIs in Javascript
- Node.js Fundamentals
- Structure of a Hapi application
- Rendering views & templates
- Session Management

Tools:

- node & npm
- hapi.js
- inert, vision, handlebars, hapi-auth-cookie

(3) Models, Persistence & Deployment

- Introduce persistence mechanisms into an hapi application and be able to employ data validation, seeding.
- Be able to deploy a hapi application



Concepts

- Models & Schema
- Database access
- Validation of data
- Deployment

Tools

- MongoDB
- Heroku toolbelt
- Mongoose, mongoose-seeder, joi

(4) Test Driven API Development

- Be able to design, implement, test and secure a Restful API

12: APIs  <p>Exposing a programmatic interface to a service can facilitate more diverse client. These can include test clients to exercises the application, a mobile application or alternative front ends.</p>	13: TDD <p>Test Driven Development is a keystone technique in modern application development. It comprises of a range of techniques, anchored but the XUnit range of libraries and tools.</p>	14: REST  <p>APIs are said to be Restful if they adhere to a set of principles encompassing URL patterns, payload and general structure.</p>	15: Auth  <p>Securing APIs requires a different approach from session-based strategies, as we may not have a browser on the client. JSON Web Tokens is a well-established authentication mechanism for APIs.</p>
--	---	---	---

Concepts

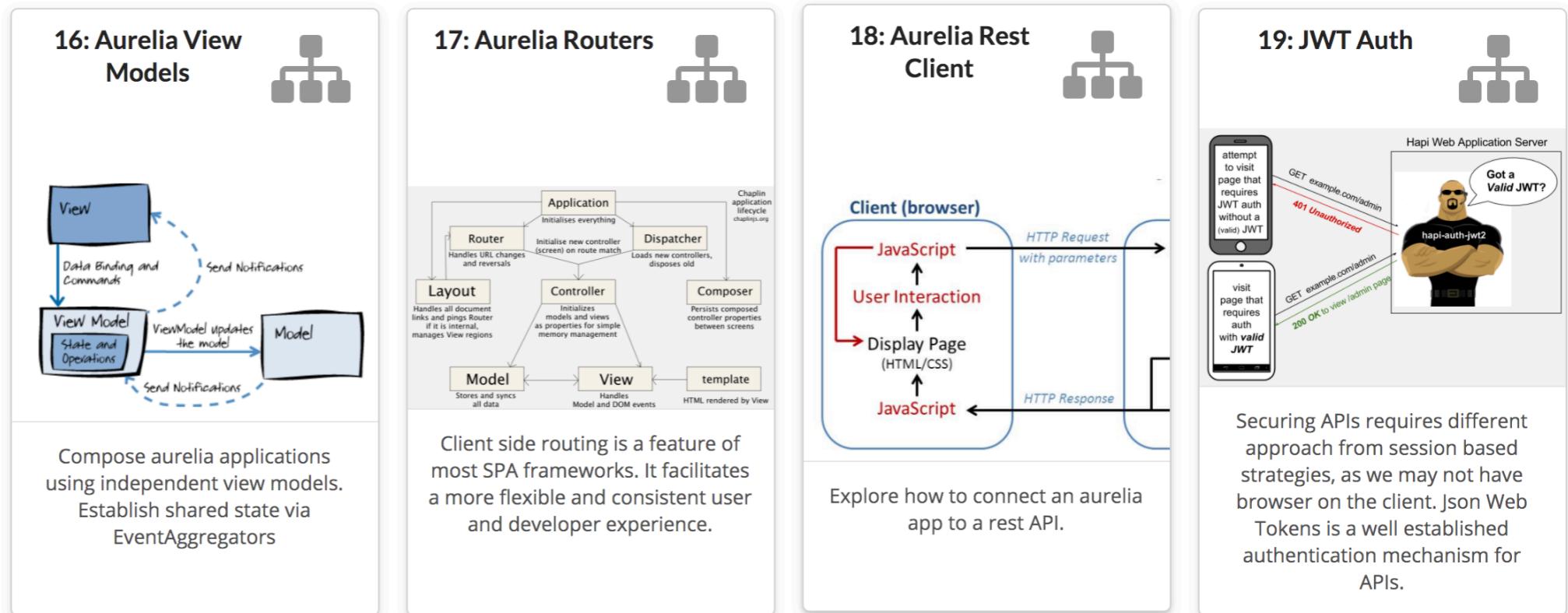
- Constructing a basic API
- TDD, Theory & Practice
- Fundamentals of REST
- Securing REST APIs

Tools:

- sync-request, mocha, chai, jsonwebtoken, hapi-auth-jwt2

(5) Single Page Applications

- Understand the SPA paradigm and be able to build a simple API driven SPA application

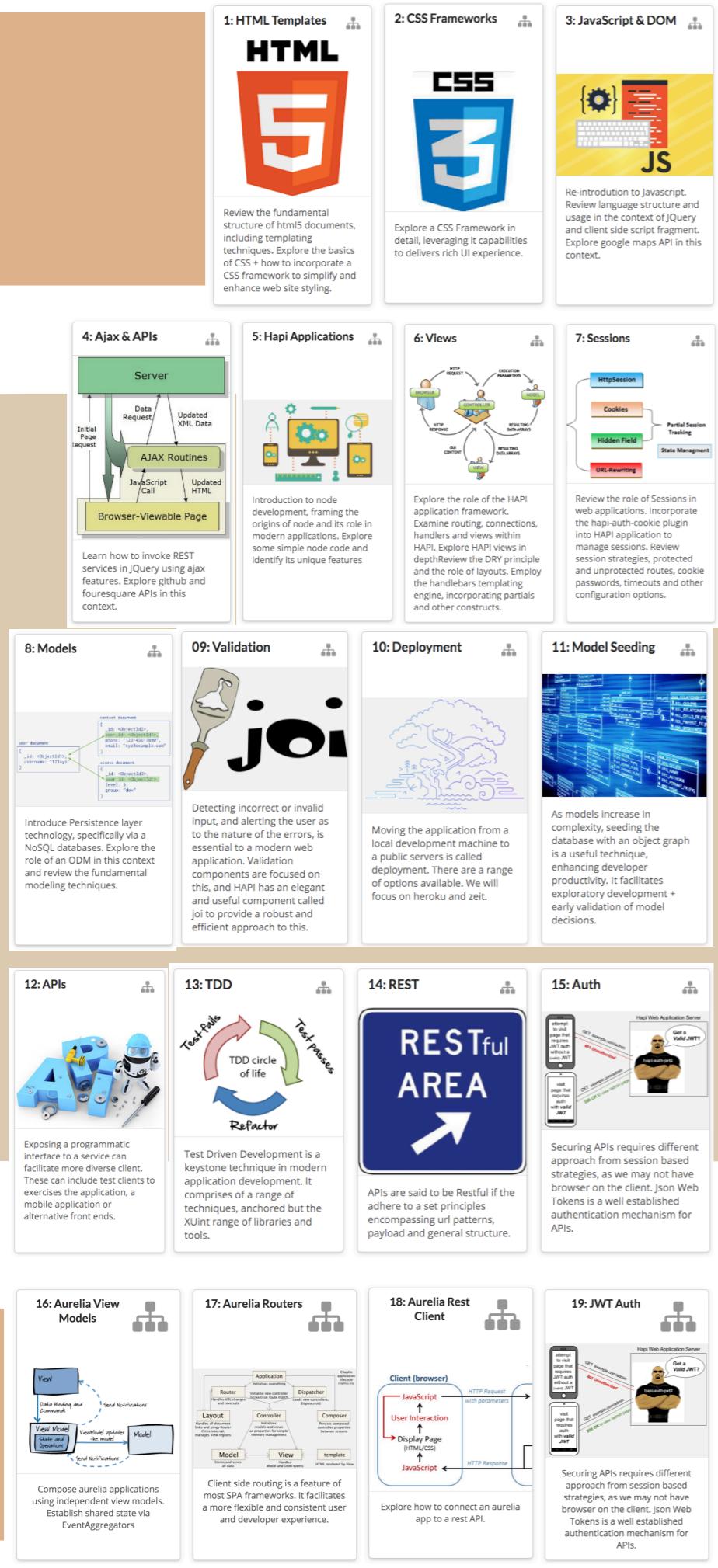


Concepts

- Typescript
- Single Page Applications
- Fundamentals of Aurelia.io
- Aurelia View Models, Routers
- JWT

Tools:

- aurelia.io
- [aurelia-cli](https://github.com/aurelia/cli)



(1) Front End Foundation

(2) APIs, Node & Hapi Applications

(3) Models, Persistence & Deployment

(4) Test Driven API Development

(5) Single Page Applications (optional)

Assessment

Single Project - submitted at
end of semester (late
December 2017)

Assessment Specification

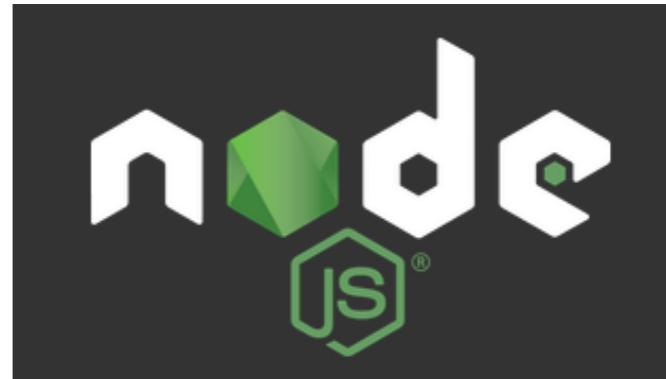
<https://wit-oth-regensburg-2017-dmas.github.io/topic-00-intro>

Assessment

The specification, submission guidelines and grading spectrum for the Assignment for this course.

Lab Requirements

- WebStorm 2016
- VSCode
- Node.js
- Mongo DB
- Robo 3T
- Heroku-cli
- + additional libraries & tools as needed in labs



A promotional graphic for WebStorm. It features the WebStorm logo (WS icon) and the text "WebStorm" in large bold letters. Below that is the tagline "The smartest JavaScript IDE". To the right is the Visual Studio Code logo (VS icon) and the text "Visual Studio Code". A descriptive text box states: "Lightweight yet powerful IDE, perfectly equipped for complex client-side development and server-side development with Node.js".



Schedule (pre and post tuition)

Pre Tuition				
	02/10/2017	09/10/2017	16/10/2017	23/10/2017
18:45 - 20:15	00: Course Overview	01 HTML Templates	02 CSS Frameworks	03 Javascript Review I

	Post Tuition					
	06/11/2017	13/11/2017	20/11/2017	27/11/2017	04/12/2017	11/12/2017
18:45 - 20:15	18 Aurelia Intro	19 Aurelia View/ Models	20 Aurelia Routers	21 Aurelia Rest Client	22 JWT Auth	23 Aurelia & JWT

Schedule (onsite tuition)

	<i>On Site Tuition</i>								
	Sat:28h	Sun: 29th	Mon: 30st	Tue: 31st	Wed: 1st	Thursday 2nd	Friday: 3rd	Saturday 4h	
08:15-09:45									
10:00-11:30	03 Javascript Review II						09 Validation	13 TDD	
11:45-13:15	04 DOM/Ajax					10 Deployment	14 REST		
13:30:15:00	05 Node + HAPI							16 Typescript Intro	
15:15-16:45	06 Views					11 Model Seeding	17 Node + Typescript		
17:00-18:30					07 Sessions	12 APIs			
18:45-20:15					08 Models				

<https://wit-oth-regensburg-2017-dmas.github.io>

https://wit-oth-2017-dmas.git x Eamonn

Secure | https://wit-oth-2017-dmas.github.io

Building Modern Web Applications & Services

Module Home

Prepared for OTH Regensburg by Eamonn de Leistar, WIT.
Creative Commons License

1: Introduction

2: HTML & CSS Revision

3: JavaScript Review

4: DOM & JQuery

Introducing the module, assessment and schedule.

Ask Questions here:

OPEN CHAT

<https://wit-oth-regensburg-2017-dmas.github.io>

The screenshot shows a web browser window with the URL <https://wit-oth-2017-dmas.git>. The page displays a course module titled "Building Modern Web Applications & Services". The module structure is organized into two main sections: "1: Introduction" and "1: HTML & CSS Revision". Each section contains several sub-topics with icons and brief descriptions. A "Module Home" button is located at the top right of the module area. To the right of the module, there is a "good day all!" message from a user named "Prepared for OTH Regensburg Creative". Below this, a large box contains the text: "Public Chat Room: Requires account/ sign in on *https://gitter.im* first". Another box below it states: "I will drop in occasionally to answers any questions or chat in general about course". On the far right, there is a portrait photo of a man wearing glasses and a blue beanie. At the bottom right, there is a text input field with placeholder text: "Click here to type a chat message. Supports GitHub flavoured markdown.".

Building Modern Web Applications & Services

Module Home

good day all!

Prepared for OTH Regensburg Creative

1: Introduction

- 1. HTML Templates
- 2. CSS Frameworks
- 3. Javascript Orientations
- 4. Ajax & APIs
- 5. Node Applications
- 6. Views
- 7. Models
- 8. Validation
- 9. Deployment
- 10. APIs
- 11. Model Seeding
- 12. TDD
- 13. REST
- 14. Auth
- 15. SPA's

Introducing the module, assessment and schedule.

1: HTML & CSS Revision

- HTML
- CSS
- JS
- Node
- Views
- Models
- Validation
- Deployment
- APIs
- Model Seeding
- TDD
- REST
- Auth
- SPA's

STRUCTURAL LAYER

PRESENTATION LAYER

Review of the fundamentals of HTML Templating + the Semantic UI CSS Framework

Public Chat Room:

Requires account/
sign in on *https://gitter.im* first

I will drop in occasionally to answers any questions or chat in general about course

Click here to type a chat message. Supports GitHub flavoured markdown.

2016 course is still online.

However,
assessment and
content will be
refreshed

The screenshot shows a web browser displaying the course website at <https://edeleastar.github.io/oth-regensburg>. The page title is "Building Modern Web Applications & Services". The top navigation bar includes links for "All Slides" and "All Labs". Below the title, a section titled "Top Level Topics" lists 11 topics, each with a thumbnail image and a brief description:

- 0: Assignments**: Calendar and specifications for the assessments on the course.
- 1: HTML Templates**: Review the fundamental structure of HTML5 documents, including templating techniques. Explore the basics of CSS + how to incorporate a CSS framework to simplify and enhance web site styling.
- 2: CSS Frameworks**: Explore a CSS Framework in detail, leveraging its capabilities to deliver rich UI experience.
- 3: JavaScript & DOM**: Re-introduction to Javascript. Review language structure and usage in the context of jQuery and client-side script fragments. Explore Google Maps API in this context.
- 4: Ajax & APIs**: Learn how to invoke REST services in jQuery using Ajax features. Explore GitHub and Foursquare APIs in this context.
- 5: Hapi Applications**: Introduction to node development, framing the origins of node and its role in modern applications. Explore some simple node code and identify its unique features.
- 6: Views**: Explore the role of the HAPI application framework. Examine routing, connections, handlers and views within HAPI. Explore HAPI views in depth. Review the DRY principle and the role of layouts. Employ the Handlebars templating engine, incorporating partials and other constructs.
- 7: Sessions**: Review the role of sessions in web applications. Incorporate the hapi-auth-cookie plugin into HAPI applications to manage sessions. Review session strategies, protected and unprotected routes, cookie passwords, timeouts and other configuration options.
- 8: Models**: Introduce Persistence layer technology, specifically via a NoSQL database. Explore the role of an ODM in this context and review the fundamental modeling.
- 9: Validation**: Detecting incorrect or invalid input, and alerting the user as to the nature of the errors, is essential to a modern web application. Validation components are and HAPI has an elegant
- 10: Deployment**: Moving the application from a local development machine to a public servers is called deployment. There are a range of options available. We will focus on heroku
- 11: Model Seeding**: As models increase in complexity, seeding the database with an object graph is a useful technique, enhancing developer productivity. It facilitates exploratory

The URL at the bottom of the page is <https://edeleastar.github.io/oth-regensburg/topic-01-html/index.html>.