

Building Modern Web Applications & Services using Node.js



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



Eamonn de Leastar
edeleastar@wit.ie

<https://edeleastar.github.io/oth-regensburg>

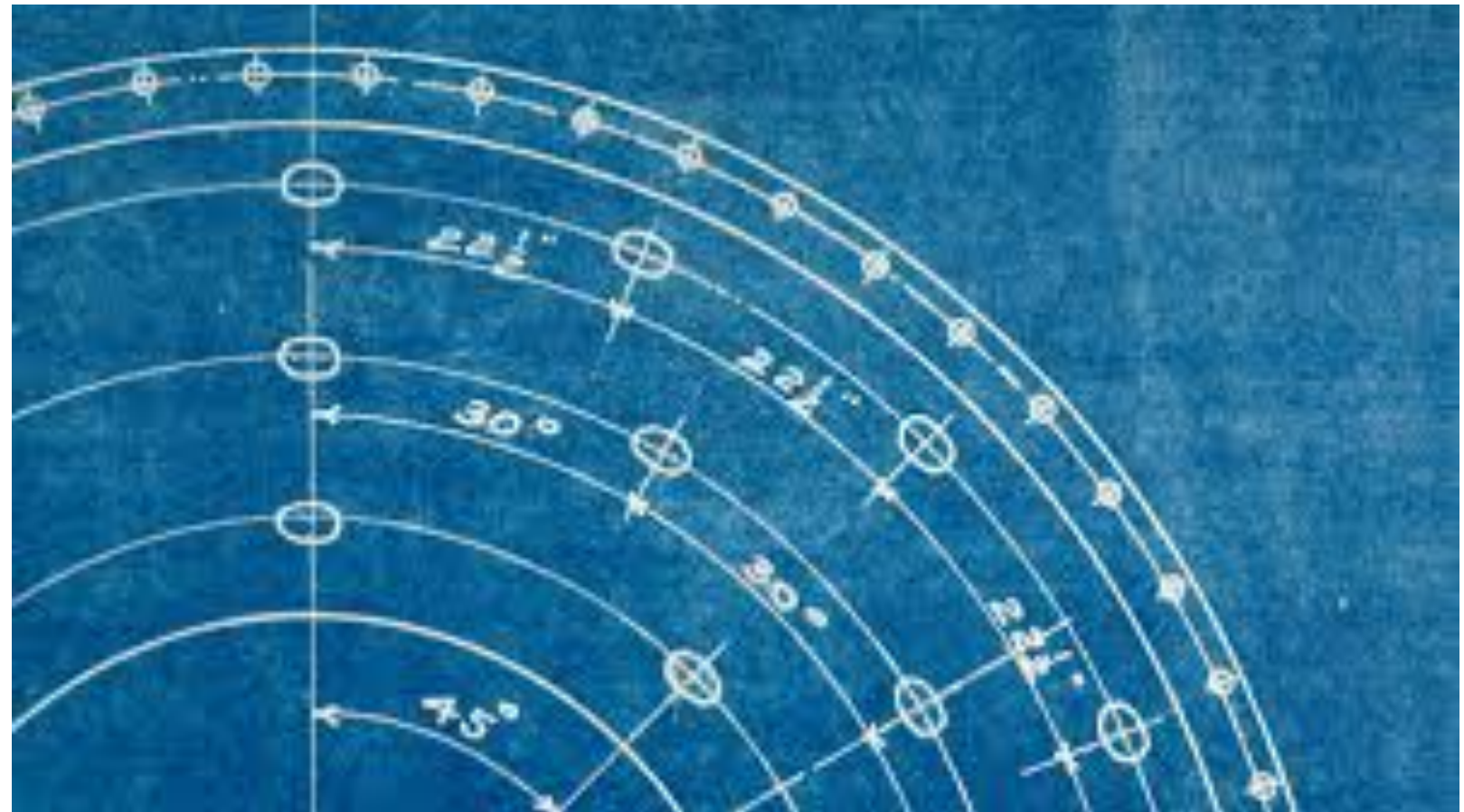
Course Mission

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

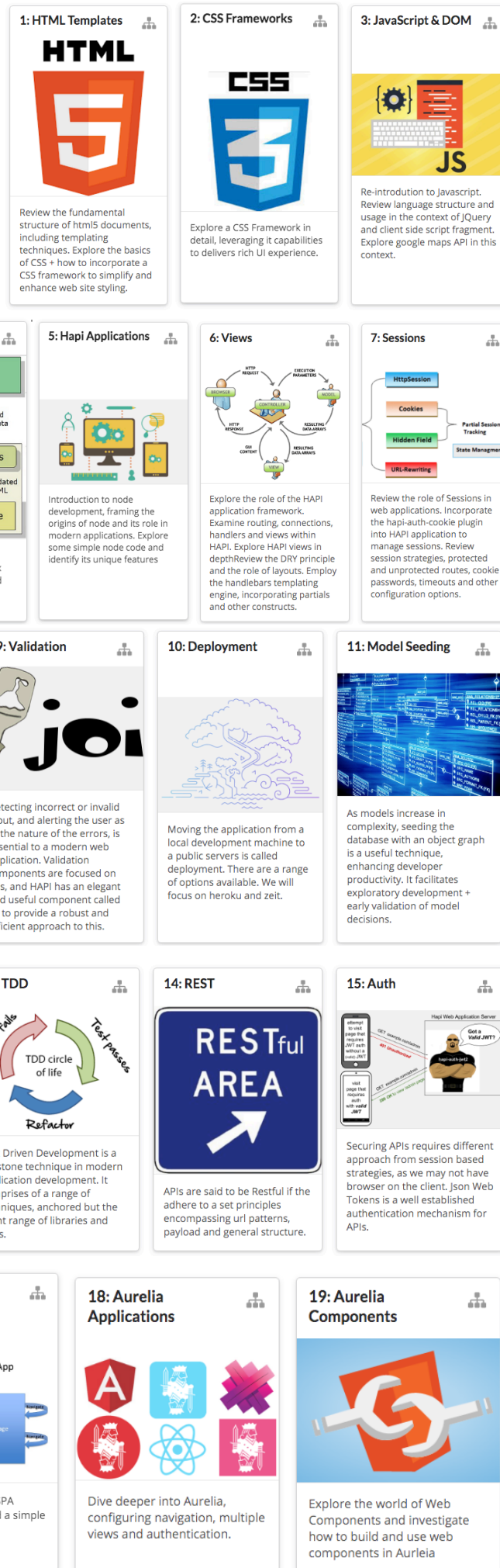


Perquisites

- 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

(2) Apis, Node & Hapi Applications

(3) Models, Persistence & Deployment

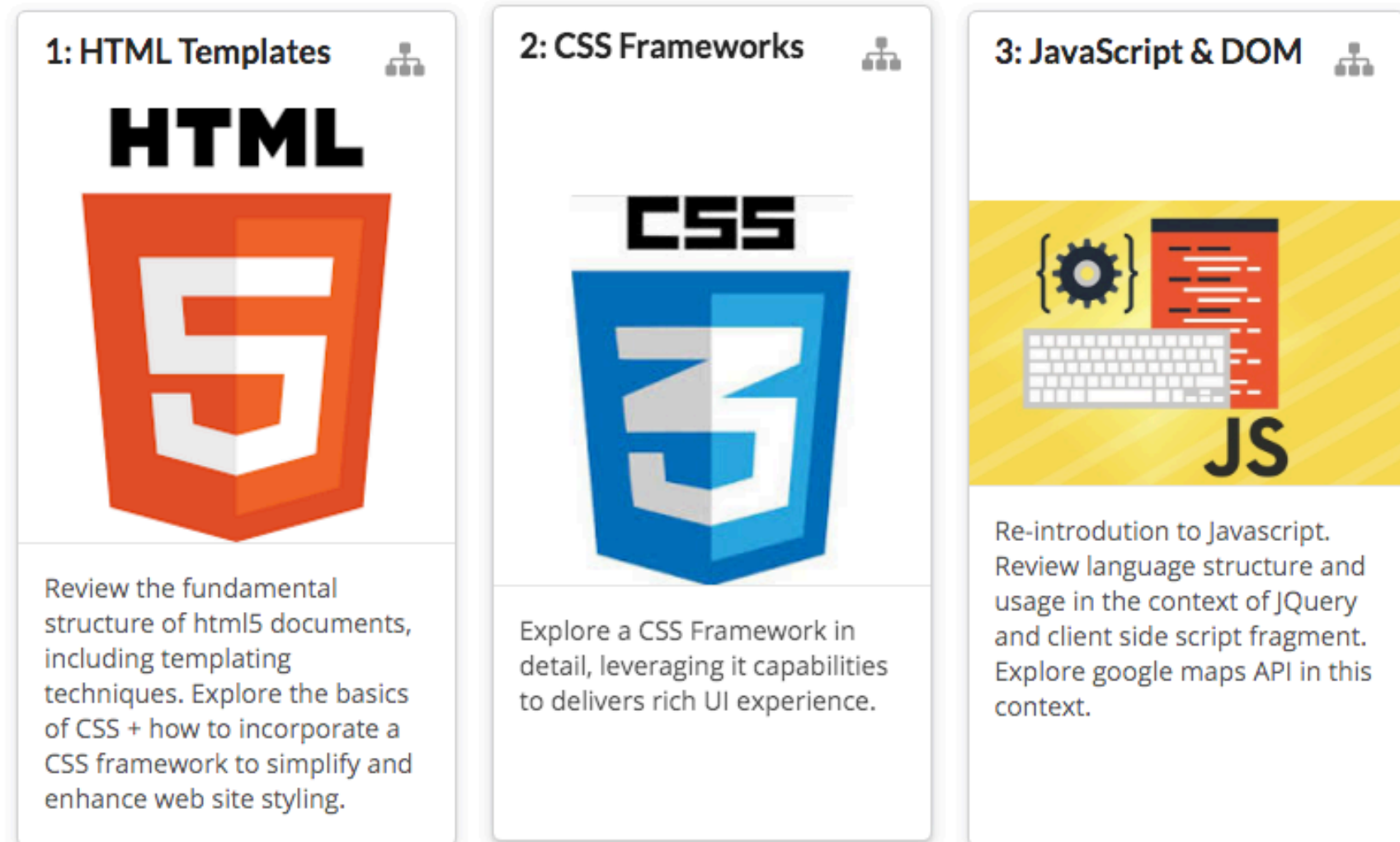
(4) Test Driven API Development

(5) Single Page Applications (optional)

(1) Front End Foundation

<https://edeleastar.github.io/oth-regensburg>

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



- Material Already Published
- Covers the prerequisites
- Students should review/complete this material prior to formal commencement of programme

Concepts

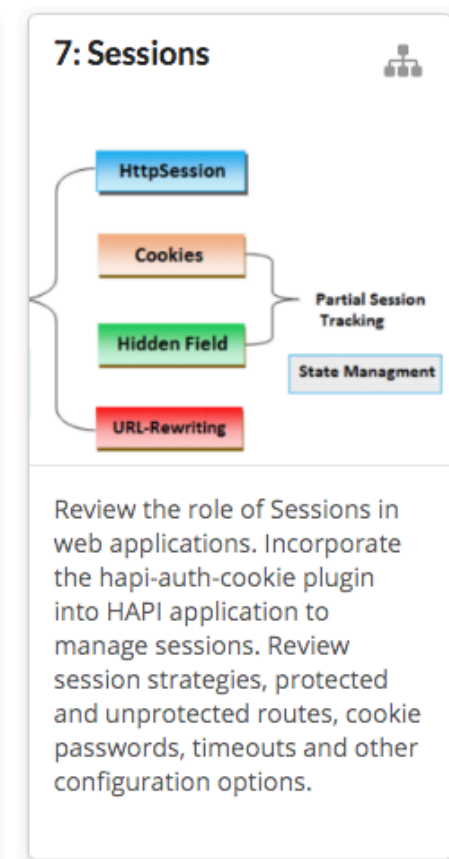
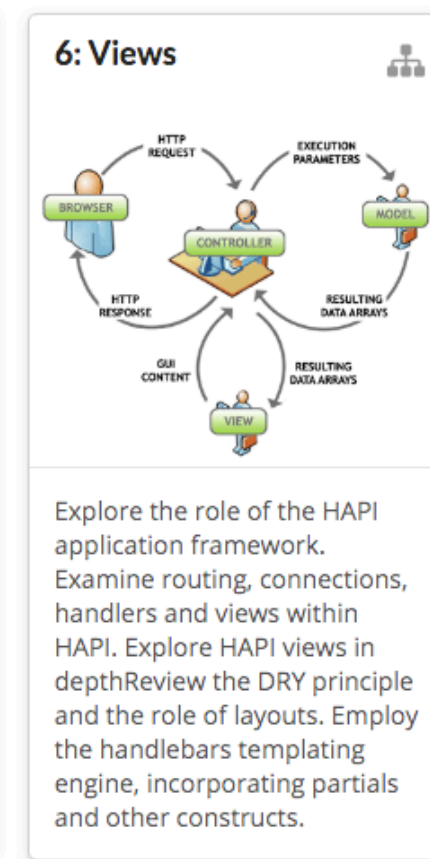
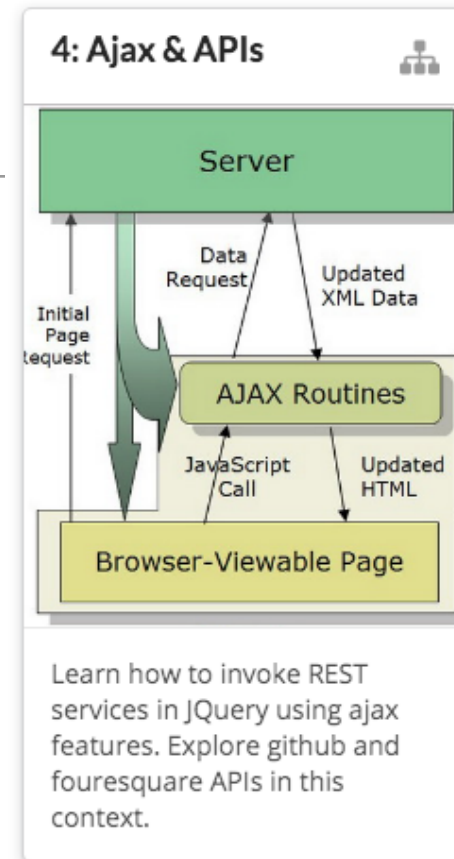
- Review Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery

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



Concepts

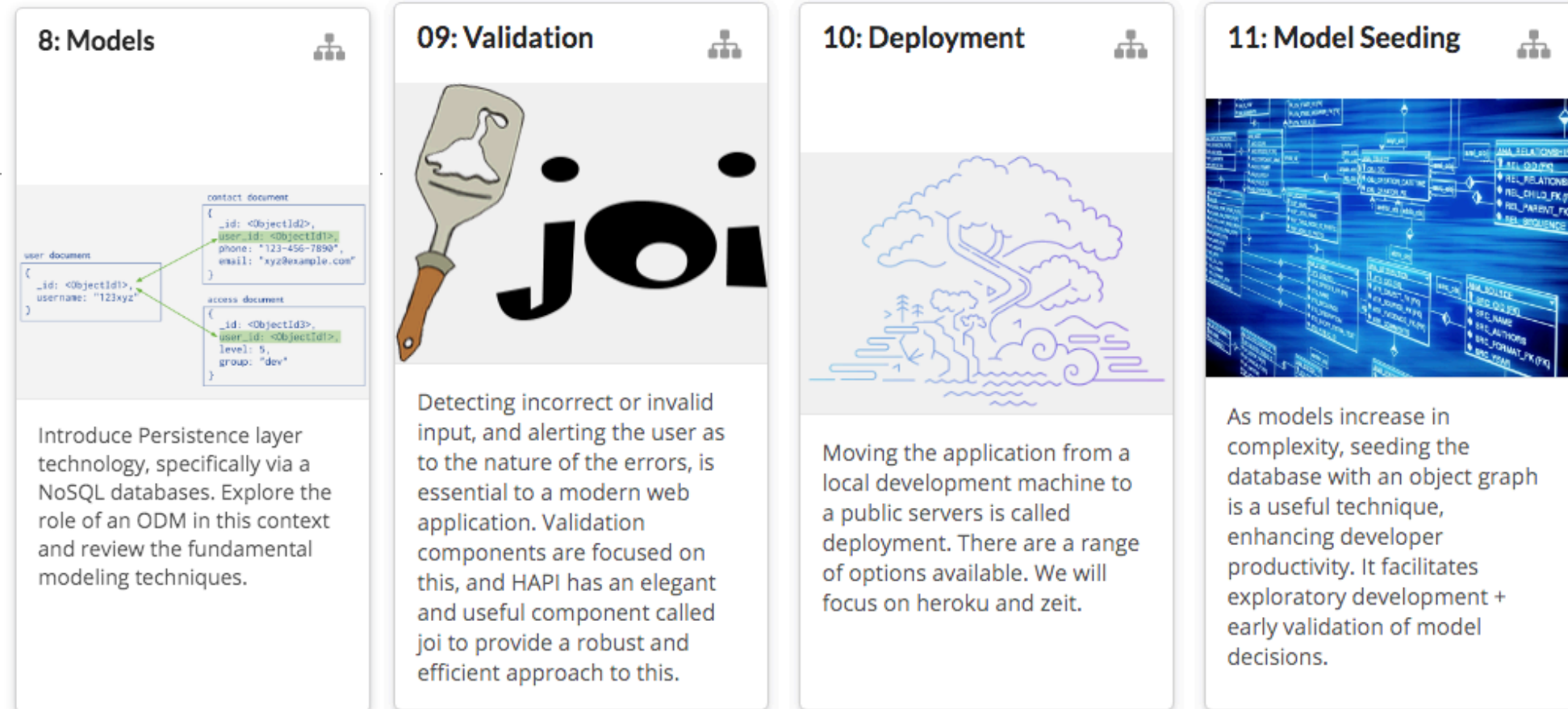
- Accessing APIs in Javascript
- Node.js Fundamentals
- Structure of a Hapi application
- Rendering views & templates
- Session Management

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

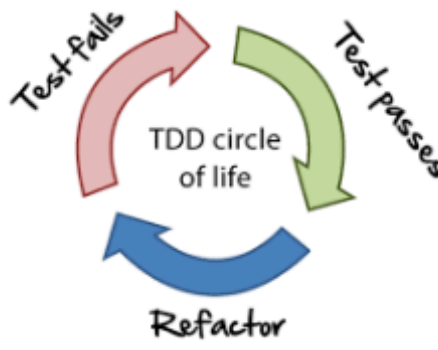
- Be able to design, implement, test and secure a Restful API

12: APIs



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.

13: TDD




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.

14: REST



APIs are said to be Restful if the adhere to a set principles encompassing url patterns, payload and general structure.

15: Auth



Securing APIs requires different approach from session based strategies, as we may not have browser on the client. Json Web Tokens is a well established authentication mechanism for APIs.

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 (optional)

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

17: Aurelia Introduction

Explore the basics of SPA applications, and build a simple SPA using Aurelia

18: Aurelia Applications

Dive deeper into Aurelia, configuring navigation, multiple views and authentication.

19: Aurelia Components

Explore the world of Web Components and investigate how to build and use web components in Aurelia

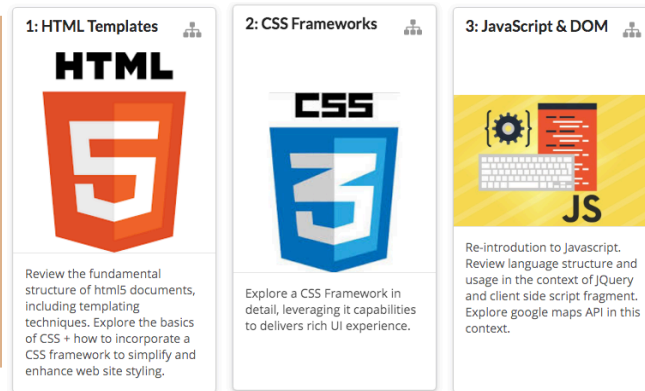
Concepts

- Single Page Applications
- Fundamentals of Aurelia.io
- Web Components
- Future of the web

Tools:

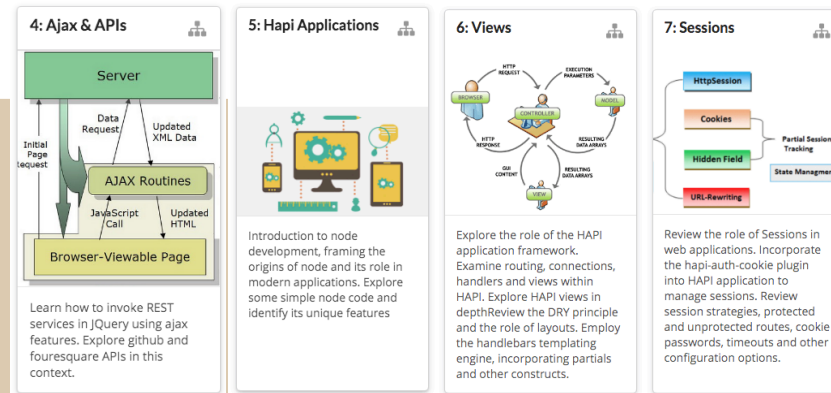
- aurelia.io
- gulp,
- aurelia-cli

before

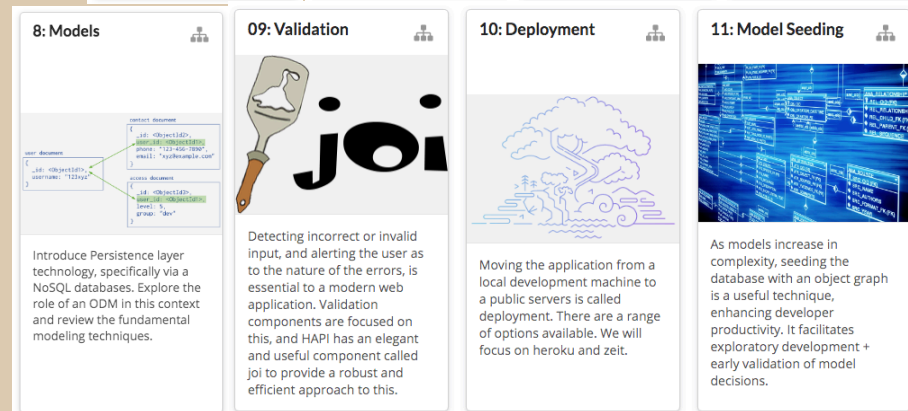


(1) Front End Foundation

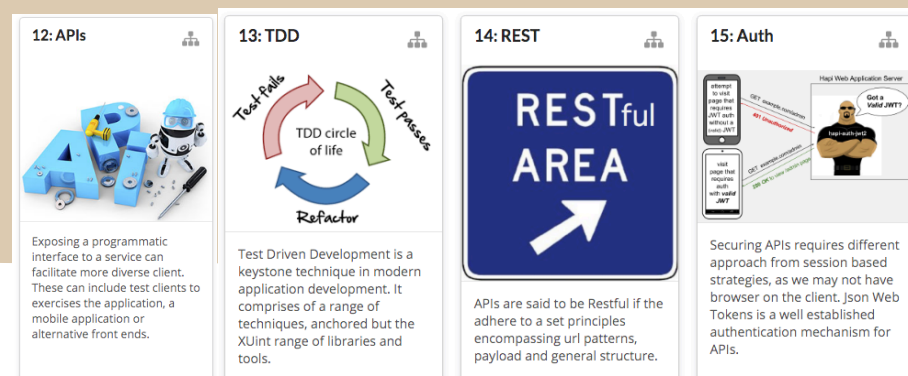
during



(2) Apis, Node & Hapi Applications

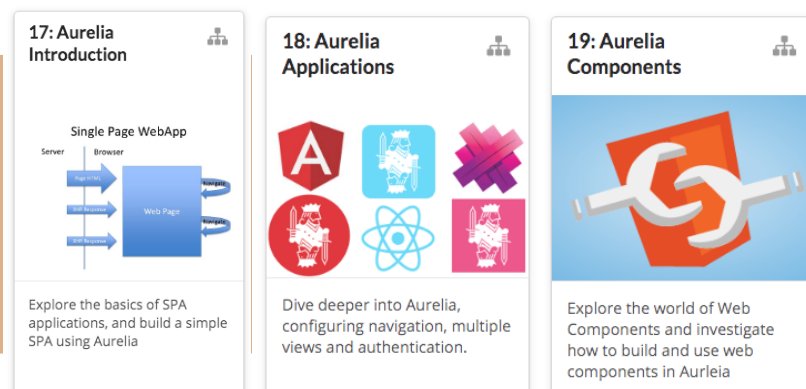


(3) Models, Persistence & Deployment



(4) Test Driven API Development

after



(5) Single Page Applications (optional)

Lab Requirements

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



Assessment Guidelines

- 2 Components:
 - Assignment: A Node Application - detailed specifications to be delivered in class
 - Research Note: A short paper (1500 words) on a specific research topic (short list to be provided)
- Deadline: End of Semester
- Marks
 - Assignment: 80%
 - Research Node: 20%

