

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>

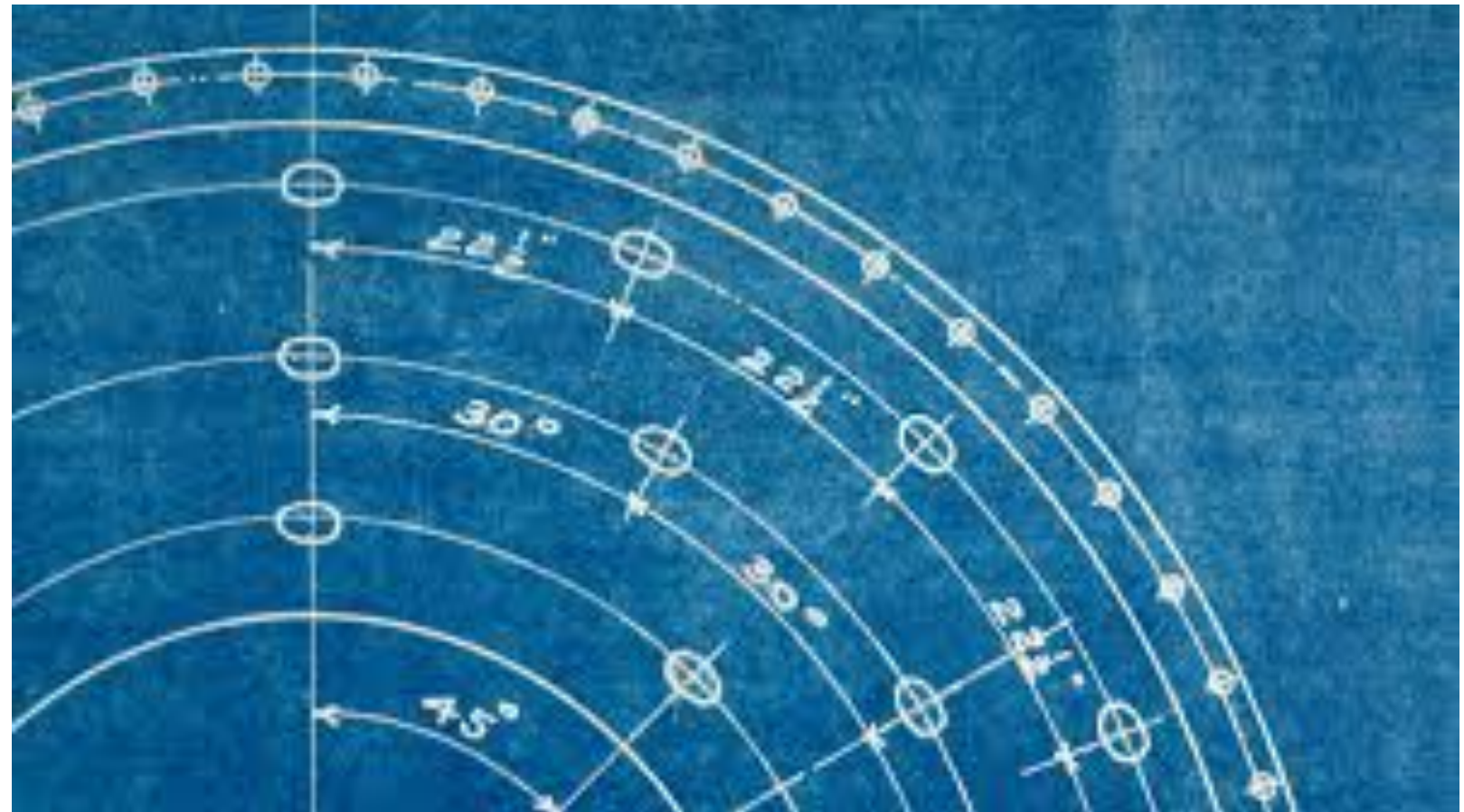
Course Mission

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
- Schedule

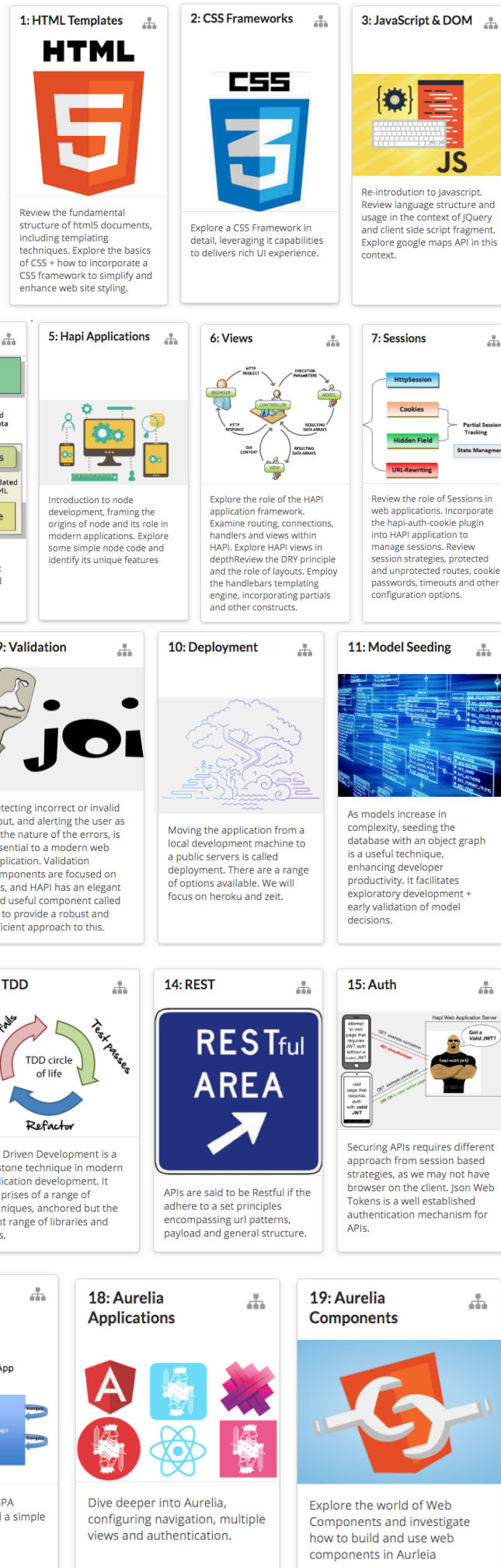


Perquisites

- 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

(2) Apis, Node & Hapi Applications

(3) Models, Persistence & Deployment

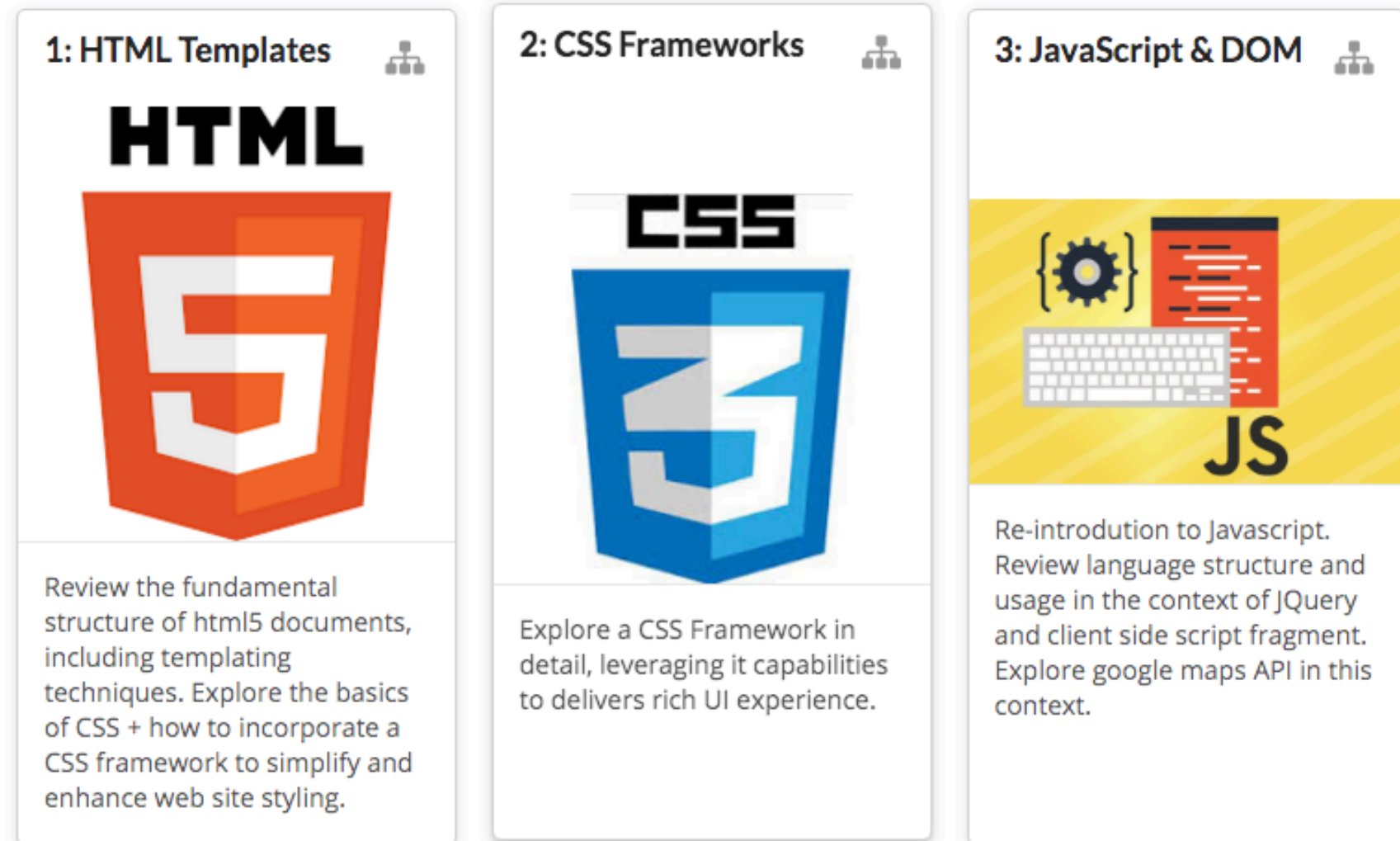
(4) Test Driven API Development

(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

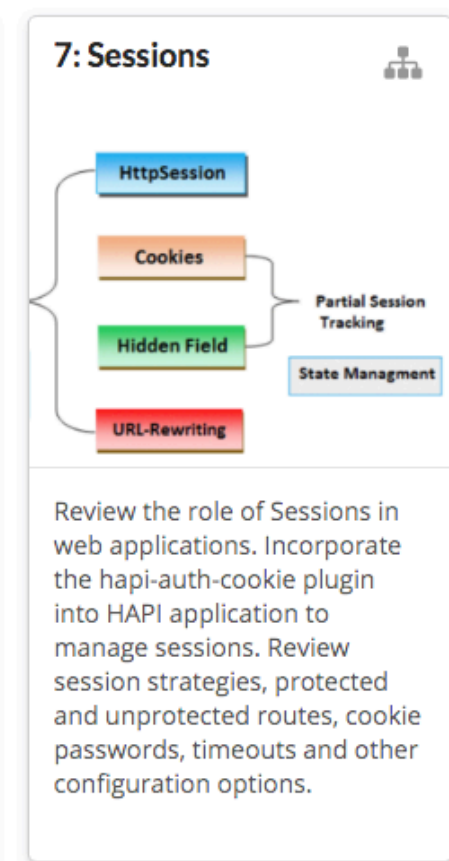
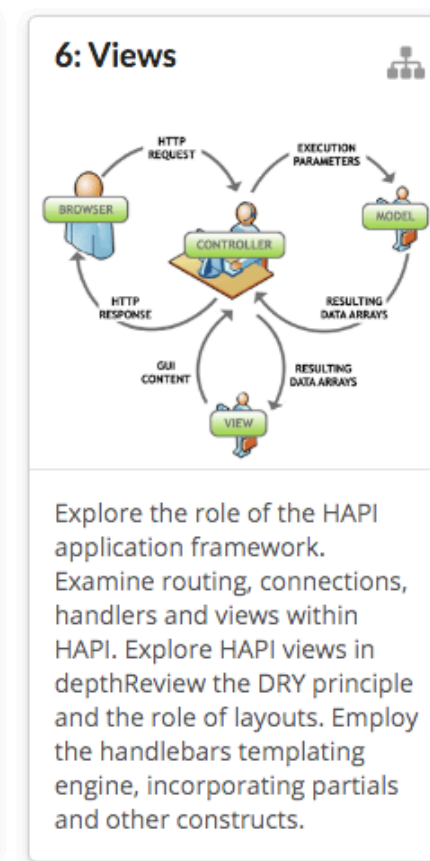
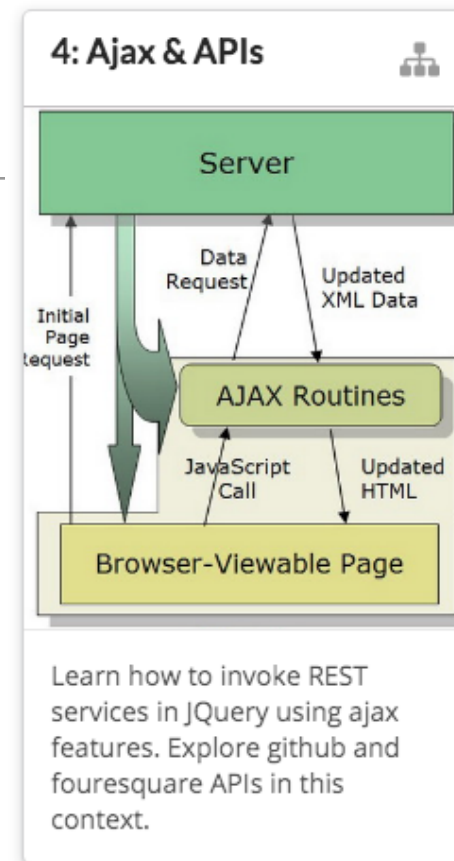
- Review Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery

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

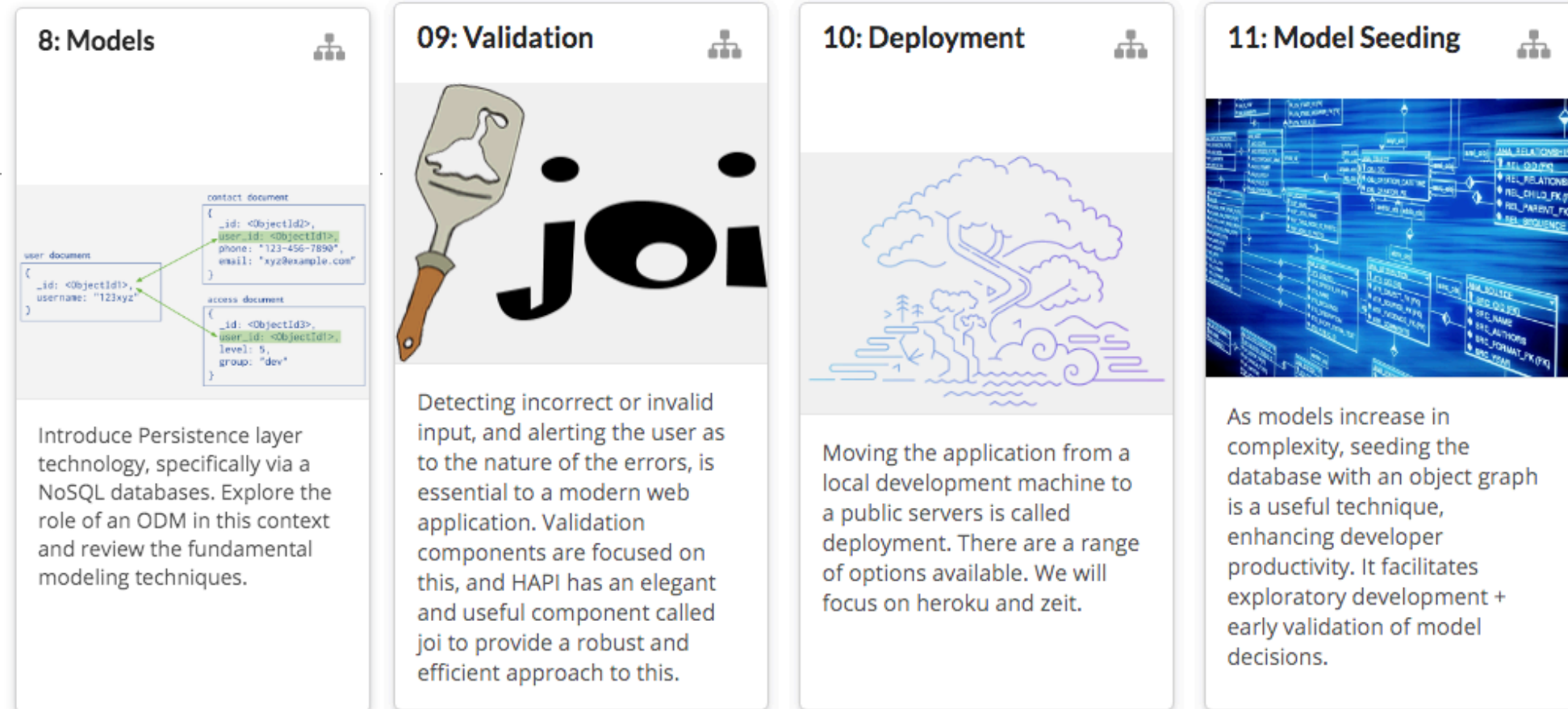
- Accessing APIs in Javascript
- Node.js Fundamentals
- Structure of a Hapi application
- Rendering views & templates
- Session Management

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

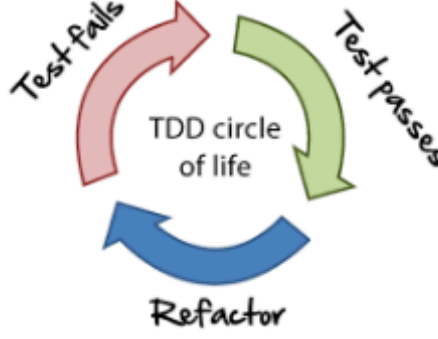
- Be able to design, implement, test and secure a Restful API

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: 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 fragment. Explore google maps API in this context.

(1) Front End Foundation


during

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 application 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 databases. Explore the role of an ODM in this context and review the fundamental modeling techniques.

09: 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 focused on this, and HAPI has an elegant and useful component called joi to provide a robust and efficient approach to this.

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 and zeit.

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 development + early validation of model decisions.

12: APIs



Exposing a programmatic interface to a service can facilitate more diverse client. These can include test clients to exercise 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.

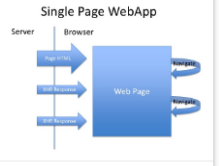
(2) Apis, Node & Hapi Applications

(3) Models, Persistence & Deployment

(4) Test Driven API Development

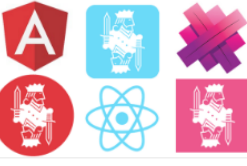
after

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

(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%



		Pre Tuition	
	01 HTML 5 Templates & Deployment	02 CSS Frameworks Semantic UI	03 Javascript
	Lab-1.1 Lab-1.2 Lab-1.3	Lab-2.1 Lab-2.2	Lab-3.1 Lab-3.2

	On Site Tuition								
	Friday:28h	Saturday: 29th	Sunday: 30th	Monday: 31st	Tuesday : 1st	Wednesday: 2nd	Thursday 3rd	Friday: 4th	Saturday 5th
08:15- 09:45							10 Deployment		
10:00- 11:30		06 Views							
11:45- 13:15		07 Sessions							
13:30: 15:00	03 Javascript Review							12 APIs	
15:15- 16:45	04 DOM/Ajax	08 Models						13 TDD	
17:00- 18:30	05 Node + HAPI							14 REST	
18:45- 20:15						09 Validation	11 Model Seeding		

Tuition			
16 Aurelia Intro	17 Aurelia Apps	18 Aurelia Compnents	
Assignment	Assignment	Assignment	
			14

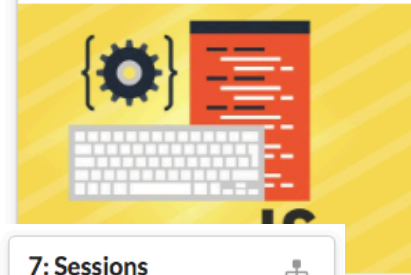
1: HTML Templates



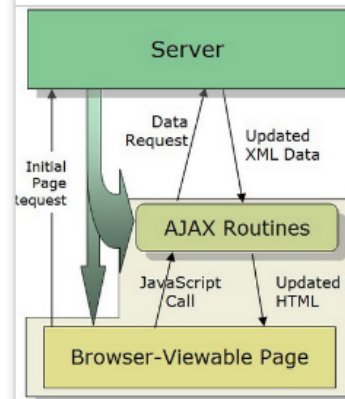
2: CSS Frameworks



3: JavaScript & DOM



4: Ajax & APIs



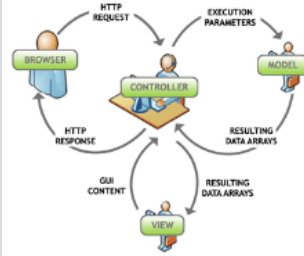
Learn how to use Ajax services in JavaScript. Explore features. Explore the context.

5: Hapi Applications



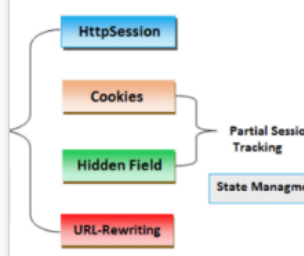
Introduction to node development, framing the origins of node and its role in modern applications. Explore

6: Views



Explore the role of the HAPI application framework. Examine routing, connections, handlers and views within

7: Sessions



Review the role of Sessions in web applications. Incorporate the hapi-auth-cookie plugin into HAPI application to

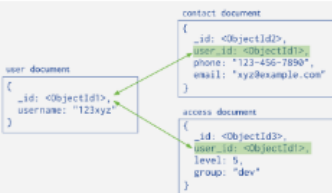
t.
and
very
rent.
this

19: Aurelia Components



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

8: Models



Introduce Pers technology, sp NoSQL database role of an ODM and review the modeling tech

09: Validation



Detecting incorrect or invalid

10: Deployment



11: Model Seeding



As models increase in

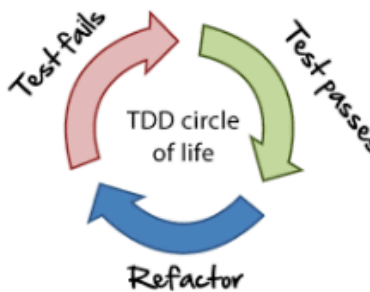
• Questions?

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.