Building Modern Web Applications & Services using Node.js

Eamonn de Leastar

edeleastar@wit.ie

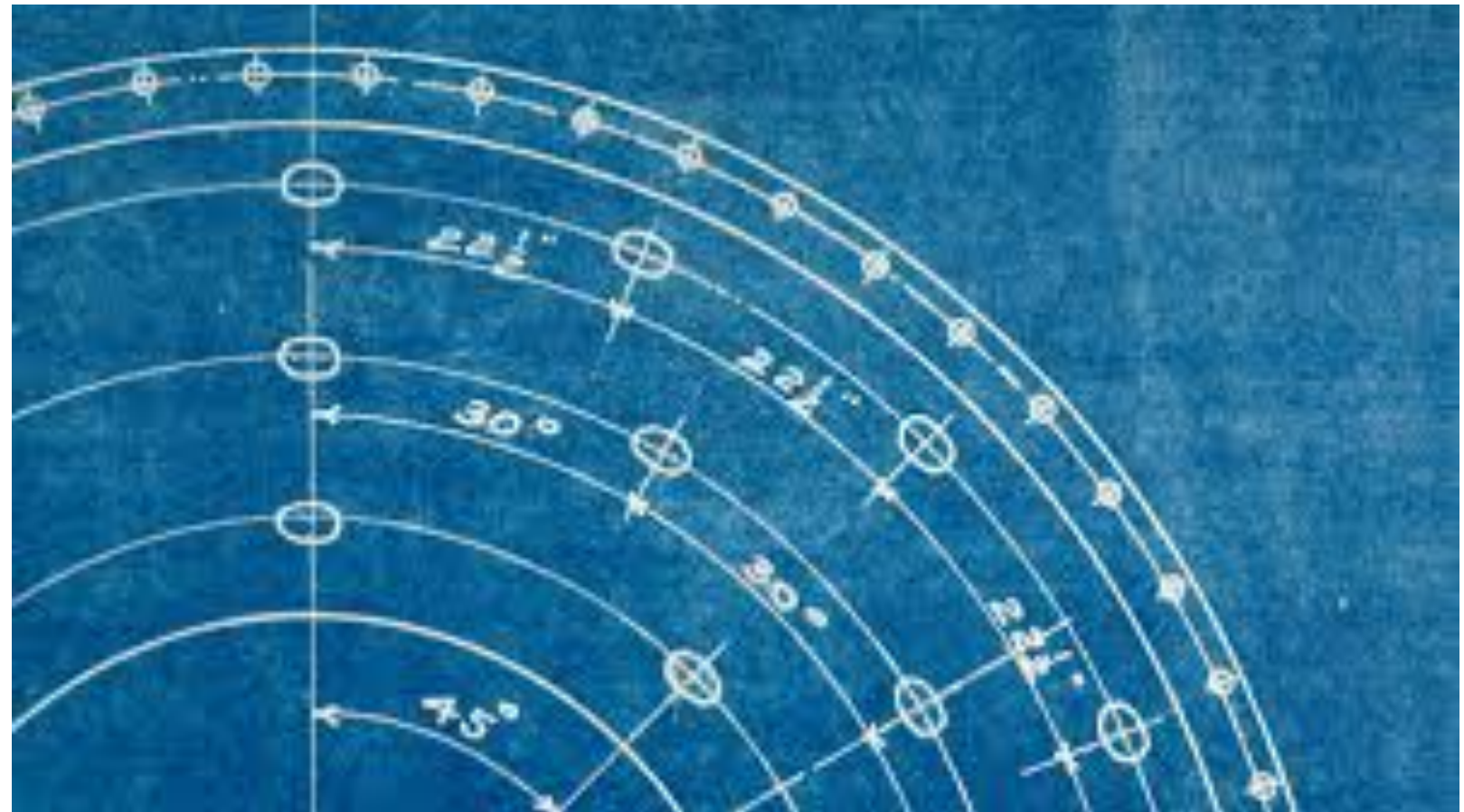wit-ictskills-2017.github.io/enterprise-web-dev

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
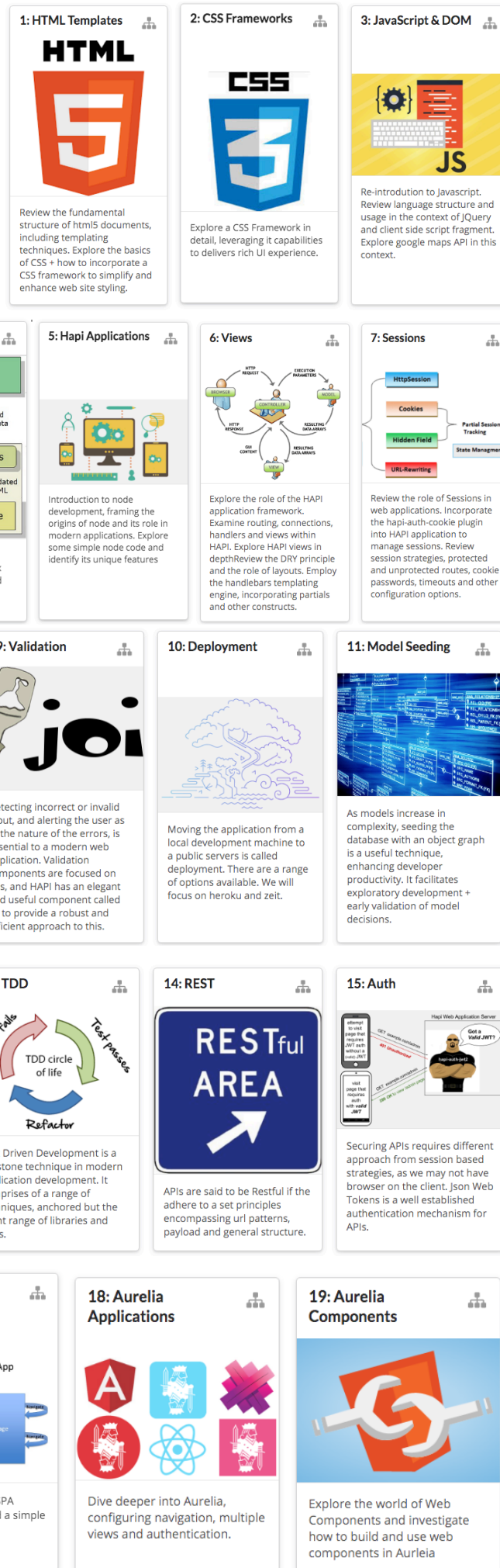
- Assessment Guidelines

- 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: 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 it capabilities to delivers 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

4: Ajax & APIs
Learn how to invoke REST services in JQuery using ajax features. Explore github and fouresquare 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 depthReview 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.

(2) Apis, Node & Hapi Applications

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.

(3) Models, Persistence & Deployment

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 XUint 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.

(4) Test Driven API Development

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 Aurlelia

(5) Single Page Applications (Typescript)

5

(1) Front End Four

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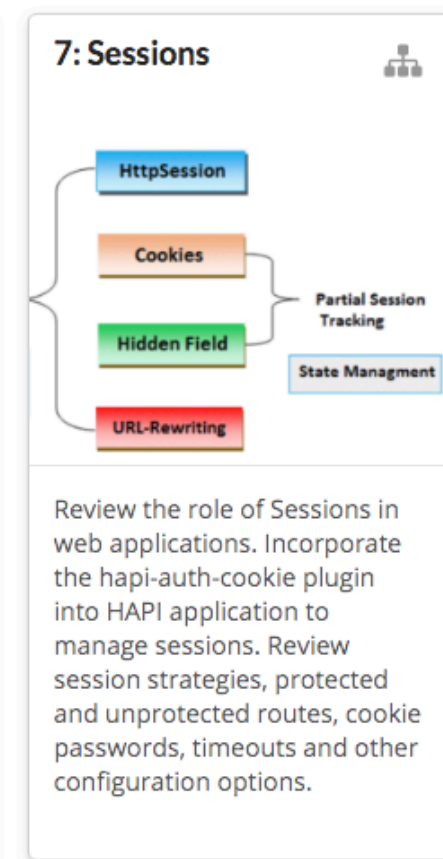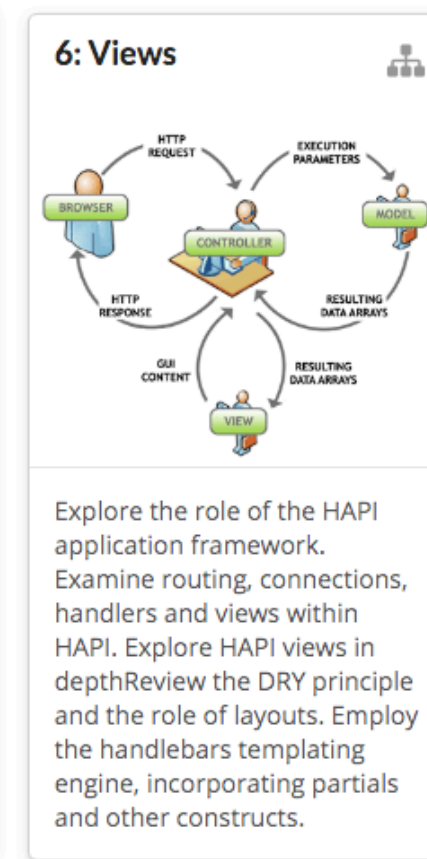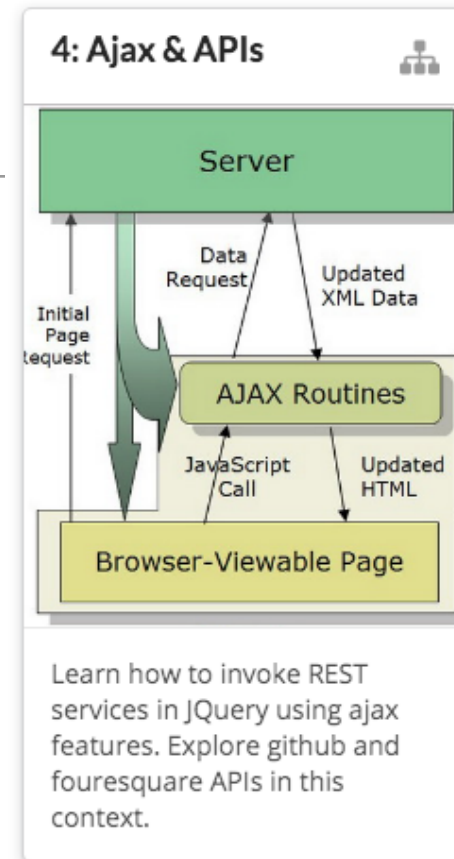
- Understand the fundamentals of Javascript & JQuery

HTML Templates

Review the construc... Introduc... F... ...sing ...use ...iques.

CSS Frameworks

Programming

Learn a broad range of programming and problem solving skills, including exc... new platforms, software tools and languages. Use these skills to... for mobile, cloud and devi... applications. Evolve a... facinating aplicatio...

Project

Building exciting IoT projects in every semester of the programme. Your projects will combine skills acquired from the other strands and enable you to build a comprehensive an compelling portfolio of IoT applications and services.

Modern web layouts are not considerably more complex and sophisticated than in the past - particularly as mobile is now considered the 'first' destination for any site. To tackle the complex issues CSS Frameworks have arisen as a convenient way to support multiple browsers and different screen sizes & resolutions.

1: JavaScript & DOM

JS

Re-introdution to Javascript. Review language structure and usage in the context of JQuery and client side script fragment. Explore google maps API in this context.

Concepts	Tools:
• Review Html + Css, focusing on templates + css frameworks	• Html5
	• Semantic-ui
• Crash course in Javascript Fundamentals	• jQuery
	• Chrome dev tools
	• DOM
• Learn basics of JQuery	• WebStorm IDE

6

(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

4: Ajax & APIs

Server

Data Request / Updated XML Data

Initial Page Request

AJAX Routines

JavaScript Call / Updated HTML

Browser-Viewable Page

Learn how to invoke REST services in JQuery using ajax features. Explore github and fouresquare 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 depthReview the DRY principle and the role of layouts. Employ the handlebars templating engine, incorporating partials and other constructs.

7: Sessions

HttpSession

Cookies

Hidden Field

URL-Rewriting

Partial Session Tracking

State Managment

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.

Concepts	Tools:
• Accessing APIs in Javascript	• node & npm
• Node.js Fundamentals	• hapi.js
• Structure of a Hapi application	• inert, vision, handlebars, hapi-auth-cookie
• Rendering views & templates	
• Session Management	

(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

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.

Concepts

- Models & Schema

- Database access

- Validation of data

- Deployment

Tools

- MongoDB

- Heroku toolbelt

- Mongoose, mongoose-seeder, joi

8

(4) Test Driven API Development
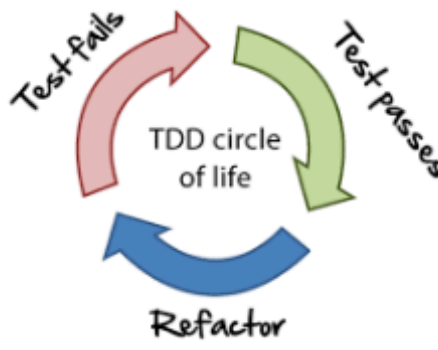
- Be able to design, implement, test and secure a Restful API

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 XUint 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

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

17: Aurelia Introduction

Single Page WebApp

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 Aurleia

Concepts

- Typescript

- Single Page Applications

- Fundamentals of Aurelia.io

- Web Components

- Future of the web

Tools:

- aurelia.io

- gulp,

- aurelia-cli

10

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 it capabilities to delivers rich UI experience.

3: JavaScript & DOM

Re-introdution 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

4: Ajax & APIs
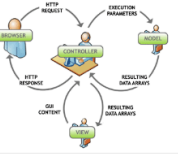
Learn how to invoke REST services in JQuery using ajax features. Explore github and foursquare APIs in this context.

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 depthReview 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.

(2) Apis, Node & Hapi Applications

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.
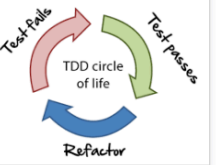
(3) Models, Persistence & Deployment

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 XUint 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.
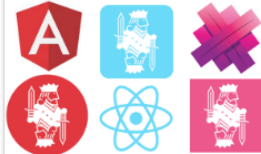
(4) Test Driven API Development

17: Aurelia Introduction

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

18: Aurelia Applications

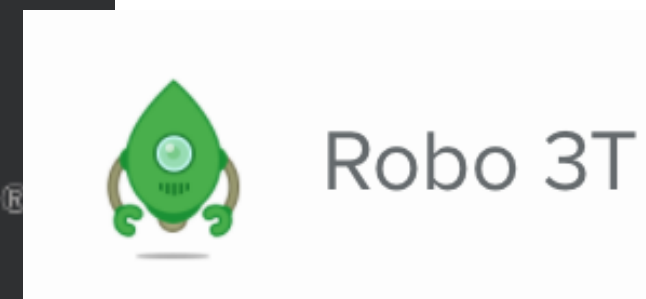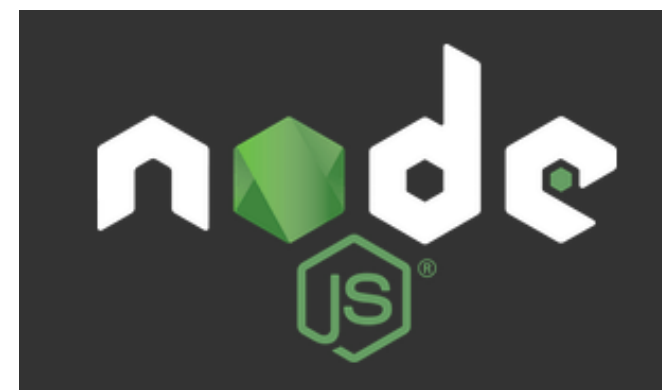Dive deeper into Aurelia, configuring navigation, multiple views and authentication.

19: Aurelia Components

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

(5) Single Page Applications (optional)

11

Lab Requirements

- WebStorm 2016

- VSCode

- Node.js

- Mongo DB

- Robo 3T

- Heroku-cli

- + additional libraries & tools as needed in labs

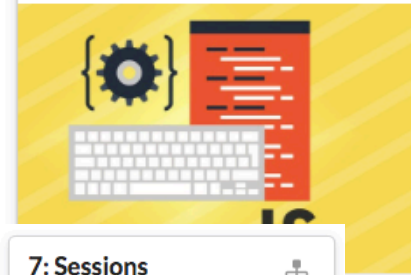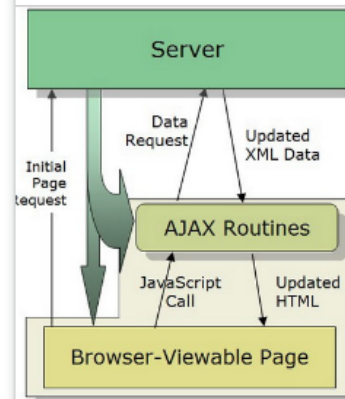**1: HTML Templates**

2: CSS Frameworks

3: JavaScript & DOM

4: Ajax & APIs

Server

Data Request / Updated XML Data

Initial Page Request

AJAX Routines

JavaScript Call / Updated HTML

Browser-Viewable Page

Learn how t
services in J
features. Ex
foursquare
context.

5: Hapi Applications

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

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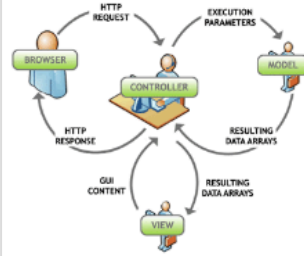
Explore the role of the HAPI application framework. Examine routing, connections, handlers and views within

7: Sessions

HttpSession

Cookies

Hidden Field

URL-Rewriting

Partial Session Tracking

State Management

Review the role of Sessions in web applications. Incorporate the hapi-auth-cookie plugin into HAPI application to
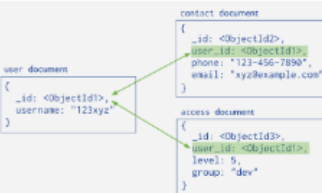
t.
and
ery
ent.
n this

19: Aurelia Components

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

8: Models

contact document
{
  _id: <ObjectId>,
  user_id: <ObjectId>,
  phone: "123-456-7890",
  email: "xyz@example.com"
}

user document
{
  _id: <ObjectId>,
  username: "123xyz"
}

access document
{
  _id: <ObjectId>,
  user_id: <ObjectId>,
  level: 5,
  group: "dev"
}

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

09: Validation

joi

Detecting incorrect or invalid
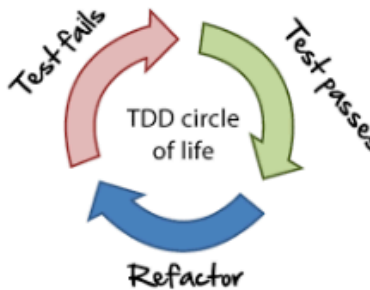
10: Deployment

11: Model Seeding

As models increase in

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 fails / Test passes

TDD circle of life

Refactor

Test Driven Development is a keystone technique in modern application development. It comprises of a range of techniques, anchored but the XUint range of libraries and tools.

14: REST

RESTful AREA

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

15: Auth

attempt to visit page that requires JWT auth without a (valid) JWT

Hapi Web Application Server

GET example.com/admin

API Unauthorized

hapi-auth-jwt2

Got a Valid JWT?

visit page that requires auth with valid JWT

GET example.com/admin

200 OK to view admin page

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.

- Questions?

13