

Building Modern Web Applications & Services using Node.js



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Eamonn de Leastar
edeleastar@wit.ie
wit-ictskills-2017.github.io/enterprise-web-dev

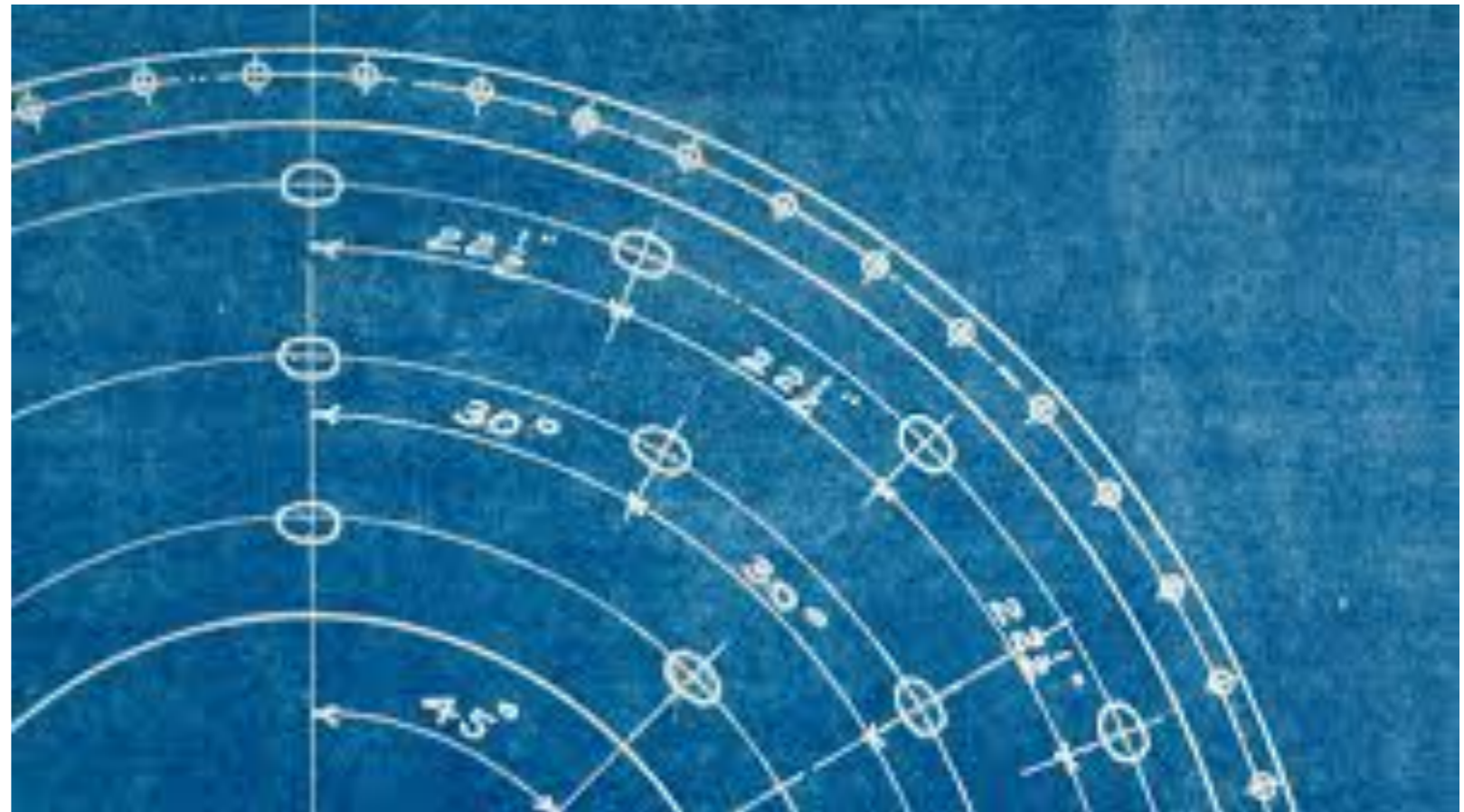
Course Mission

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

HTML Templates

Review the HTML & CSS constructs covered so far. Introduce html templating using EJS. Refactor a site to use templating techniques.

CSS Frameworks

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.

Review

Re-introduction to Javascript. Review language structure and usage.

3. DOM & JQuery

Review the nature & structure of the DOM. Basics of JQuery library

(1) Front End Foundation (+JS)

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 DRV 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.

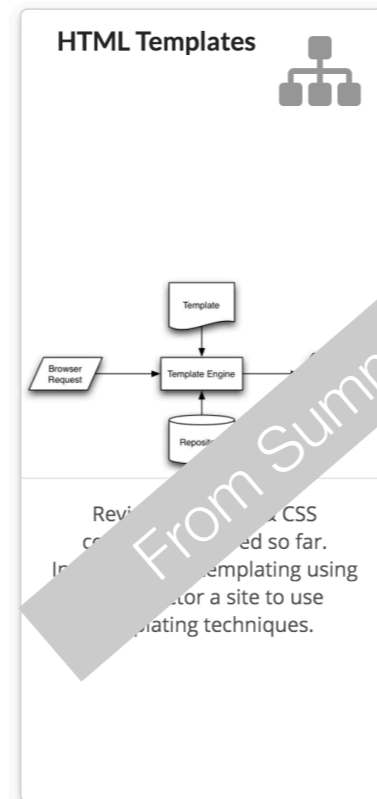
19: Aurelia Components

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

(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



CSS Frameworks

Modern web layouts are 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.

2: JavaScript Review

Re-introduction to Javascript. Review language structure and usage.

3: DOM & JQuery

Review the nature & structure of the DOM. Basics of JQuery library

Concepts

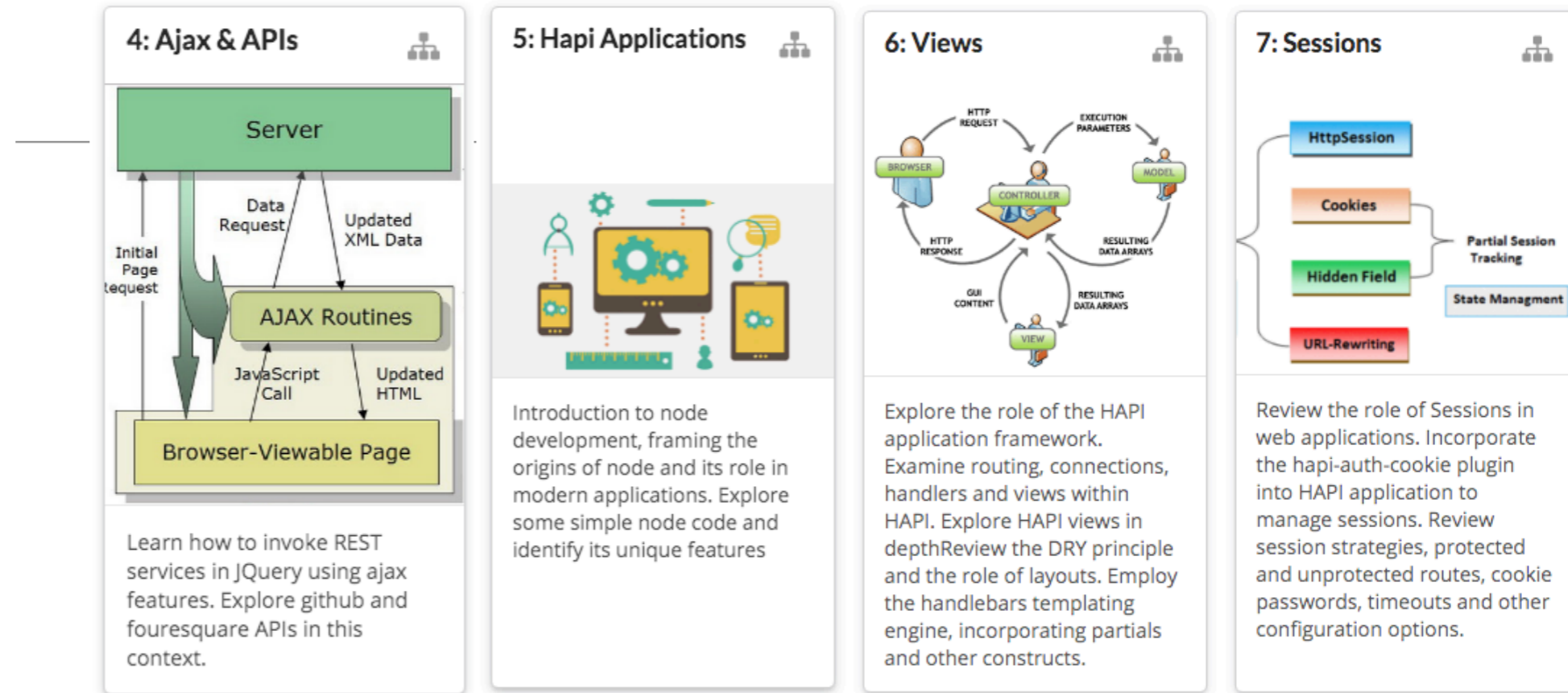
- Review Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery

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

8: Models

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

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

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

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


(4) Test Driven API Development

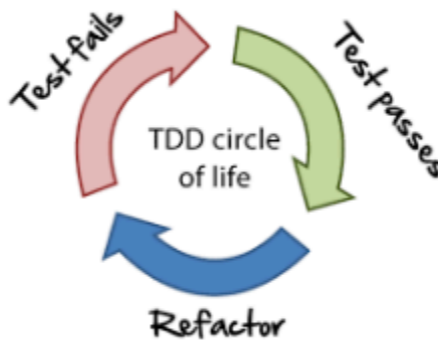
- Be able to design, implement, test and secure a Restful API

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

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

16: Aurelia View Models

Compose aurelia applications using independent view models. Establish shared state via EventAggregators

17: Aurelia Routers

Client side routing is a feature of most SPA frameworks. It facilitates a more flexible and consistent user and developer experience.

18: Aurelia Rest Client

Explore how to connect an aurelia app to a rest API.


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


- ### Tools:
- aurelia.io
 - aurelia-cli

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 delivers rich UI experience.

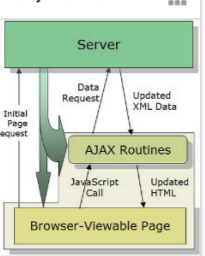
3: JavaScript & DOM



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 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 DRV 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.

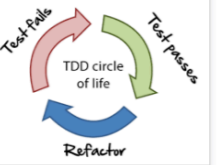
(3) Models, Persistence & Deployment

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.

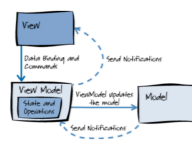
15: Auth



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.

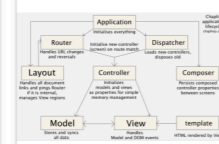
(4) Test Driven API Development

16: Aurelia View Models



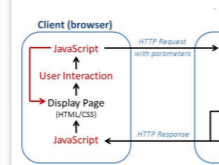
Compose aurelia applications using independent view models. Establish shared state via EventAggregators

17: Aurelia Routers




Client side routing is a feature of most SPA frameworks. It facilitates a more flexible and consistent user and developer experience.

18: Aurelia Rest Client



Explore how to connect an aurelia app to a rest API.

19: JWT Auth

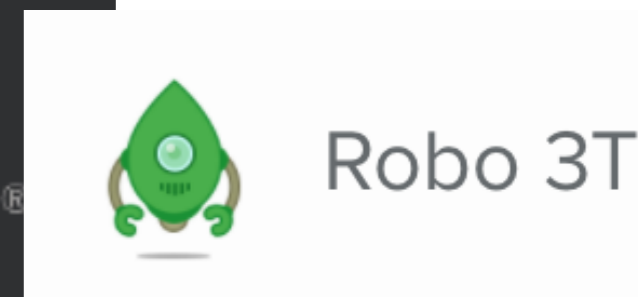
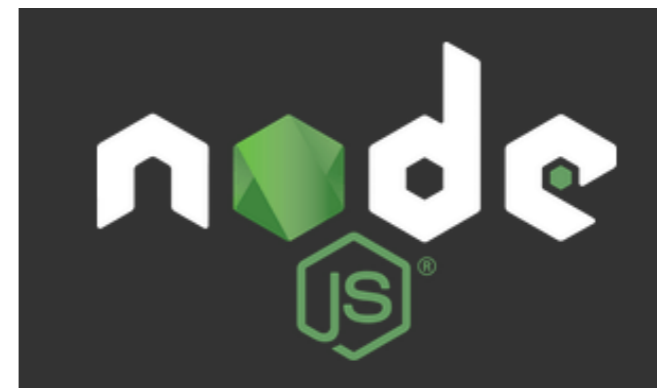


Securing APIs requires different approach from session based strategies, as we may not have browser on the client. Json Web Tokens is a well established authentication mechanism for APIs.

(5) Single Page Applications (optional)

Lab Requirements

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



HTML Templates

CSS Frameworks

2: JavaScript Review

3: DOM & JQuery

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

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

17: Aurelia Routers

Chaplin application lifecycle chaplins.org

18: Aurelia Res Client

Explore how to connect an app to a res

8: Models

09: Validation

Detecting incorrect or invalid

10: Deployment

11: Model Seeding

As models increase in

As a feature of .s. It facilitates consistent user experience.

12: APIs

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

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.

• Questions?