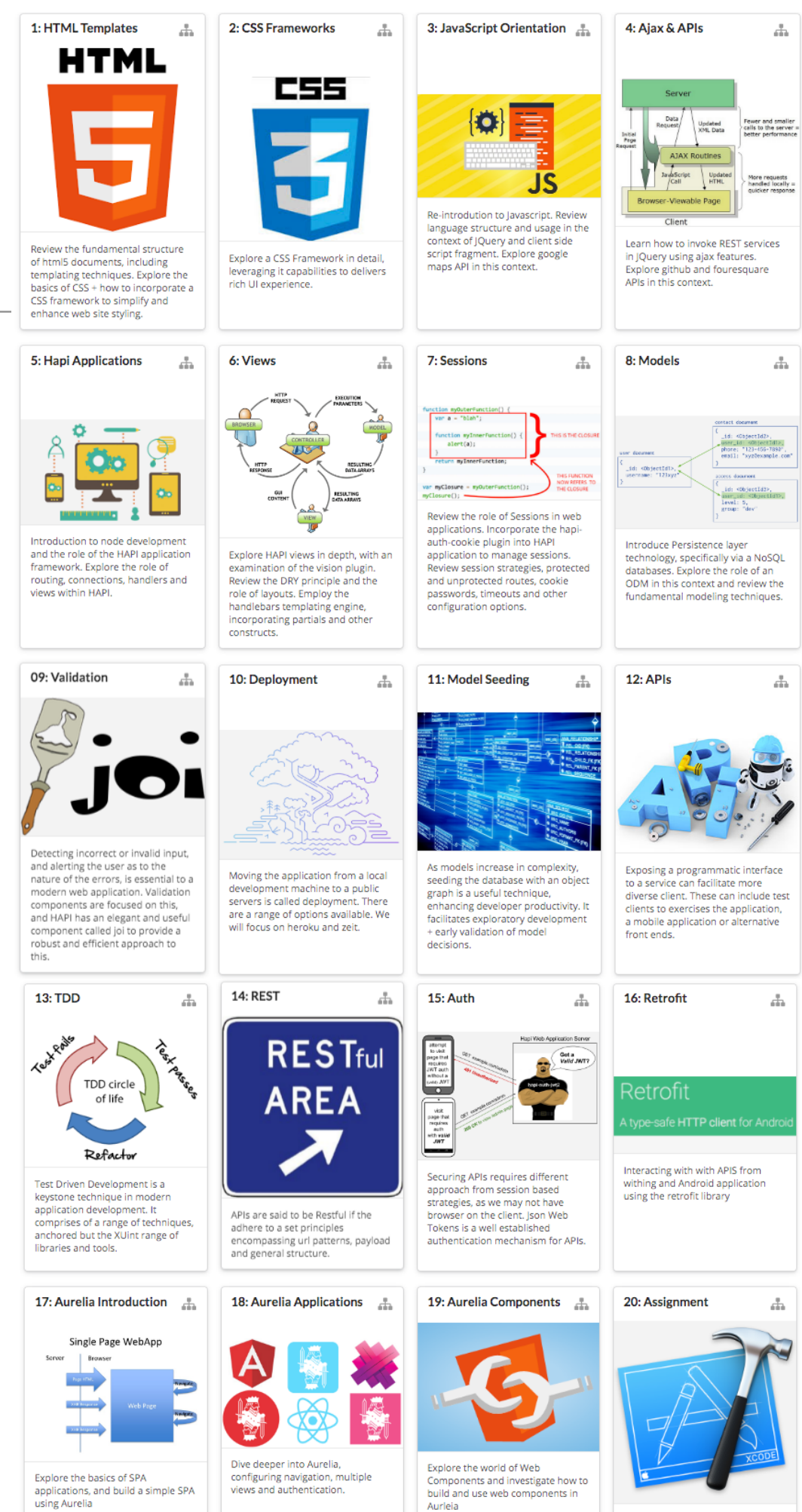


Enterprise Web Development

House of Cards

- Each topic builds on previous topics
- Objective:
 - Transfer a set of foundation skills to enable you to design, build, secure, test and deploy a modern web application + API.*



Front End Foundation


Core Node & Hapi

Building & Consuming APIs

Test Driven API Development


Single Page Applications

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



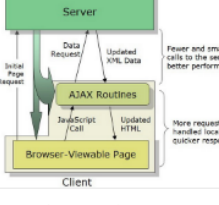
Explore a CSS Framework in detail, leveraging its capabilities to deliver rich UI experience.

3: JavaScript Orientation



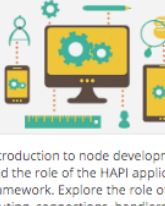
Re-introduction to JavaScript. Review language structure and usage in the context of jQuery and client side script fragment. Explore google maps API in this context.

4: Ajax & APIs



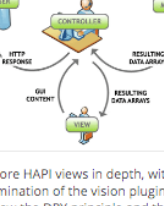
Learn how to invoke REST services in jQuery using ajax features. Explore github and foursquare APIs in this context.

5: Hapi Applications



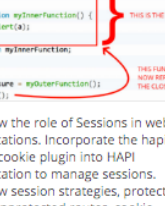
Introduction to node development and the role of the HAPI application framework. Explore the role of routing, connections, handlers and views within HAPI.

6: Views



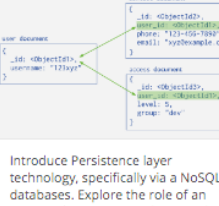
Explore HAPI views in depth, with an examination of the vision plugin. Review the DRY principle and the role of layouts. Employ the handlebars templating engine, incorporating partials and other constructs.

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



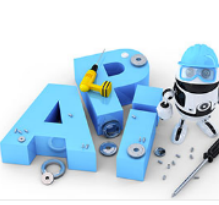
Moving the application from a local development machine to a public servers is called deployment. There are a range of options available. We will focus on heroku and zeit.

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



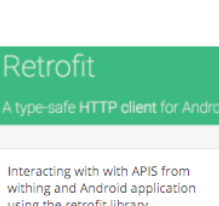
APIs are said to be Restful if they adhere to a set principles encompassing url patterns, payload and general structure.

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.

16: Retrofit




Interacting with APIs from within an Android application using the retrofit library

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

20: Assignment



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



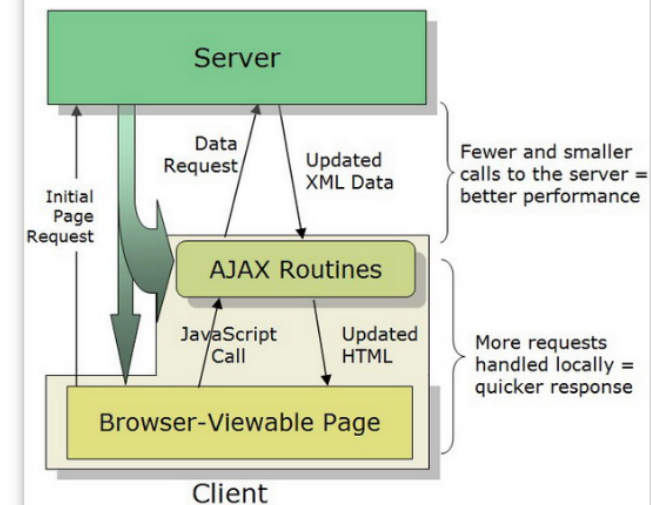
Explore a CSS Framework in detail, leveraging its capabilities to deliver a rich UI experience.

3: JavaScript Orientation



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.

4: Ajax & APIs



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

Front End Foundation

Concepts

- Revisit Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery
- Build Web Sites that interact with web APIs

Tools:

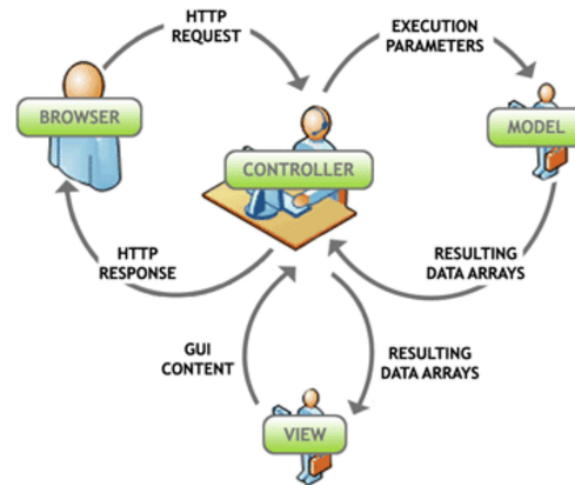
- Semantic-ui
- jQuery
- Chrome dev tools
- DOM & Ajax
- WebStorm IDE
- github & foursquare api

5: Hapi Applications



Introduction to node development and the role of the HAPI application framework. Explore the role of routing, connections, handlers and views within HAPI.

6: Views



Explore HAPI views in depth, with an examination of the vision plugin. Review the DRY principle and the role of layouts. Employ the handlebars templating engine, incorporating partials and other constructs.

7: Sessions



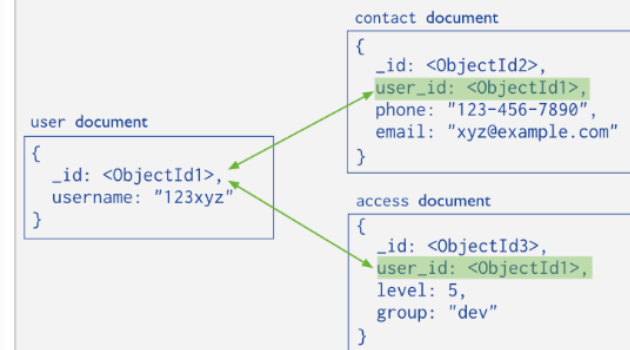
```
function myOuterFunction() {  
  var a = "blah";  
  function myInnerFunction() {  
    alert(a);  
  }  
  return myInnerFunction;  
}  
  
var myClosure = myOuterFunction();  
myClosure();
```

THIS IS THE CLOSURE

THIS FUNCTION NOW REFERS TO THE CLOSURE

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.

Core Node & HAPI

Concepts

- Node.js Fundamentals
- Structure of a Hapi application
- Rendering views & templates
- NoSQL models

Tools:

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

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



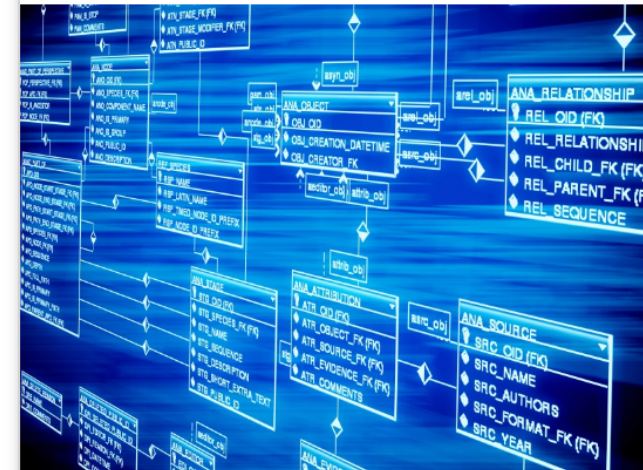
Moving the application from a local development machine to a public servers is called deployment. There are a range of options available. We will focus on heroku and zeit.

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.

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.

Building & Consuming APIs

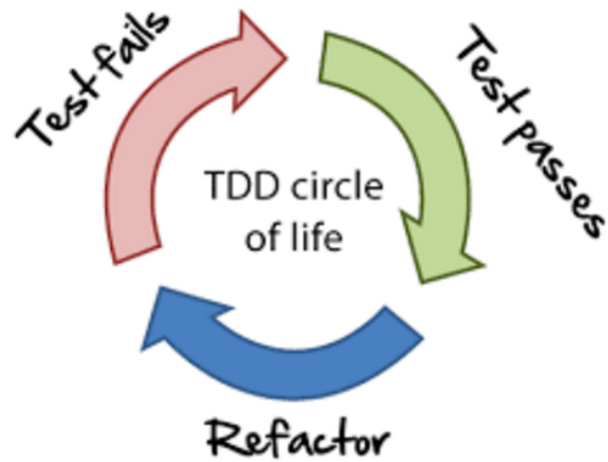
Concepts

- Rules based validation
- Application deployment
- Exposing APIs
- Seeding models for test purposes

Tools

- Heroku toolbelt
- Postman
- mongoose-seeder

13: TDD



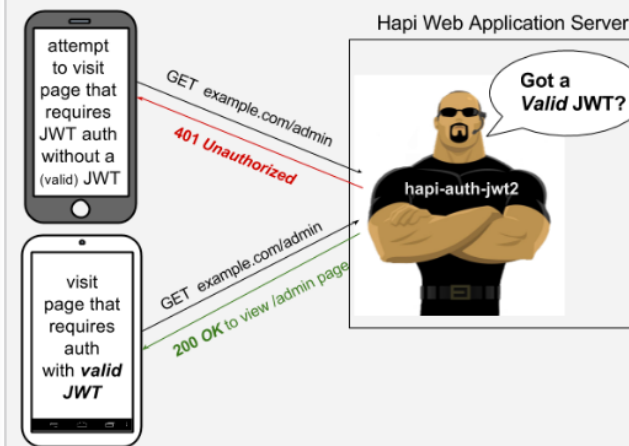
Test Driven Development is a keystone technique in modern application development. It comprises of a range of techniques, anchored but the XUnit range of libraries and tools.

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.

16: Retrofit



Retrofit

A type-safe HTTP client for Android

Interacting with with APIs from withing and Android application using the retrofit library

Test Driven API Development

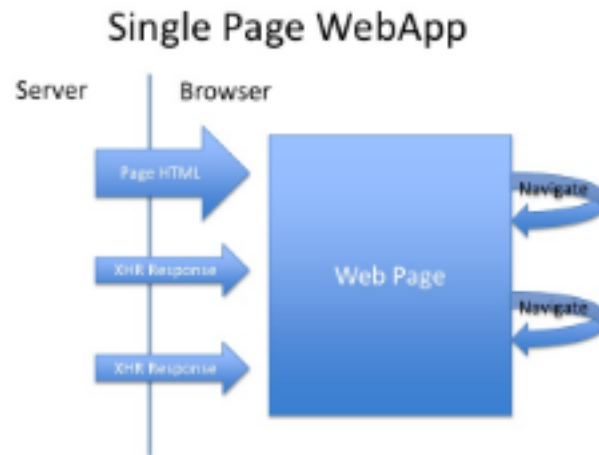
Concepts

- TDD, Theory & Practice
- Fundamentals of REST
- Securing REST APIs
- Consuming APIs from Android

Tools:

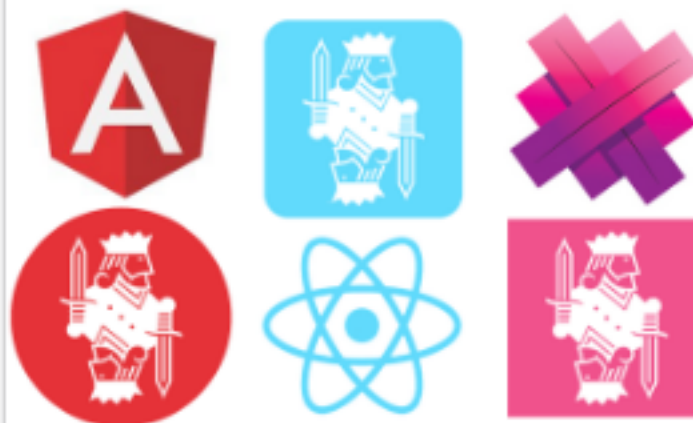
- sync-request, mocha, chai, jsonwebtoken, hapi-auth-jwt2
- Retrofit

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

Single Page Applications

Concepts

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

Tools:

- aurelia.io
- gulp, aurelia-cli
- polymer