

Building Modern Web Applications & Services using Node.js



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

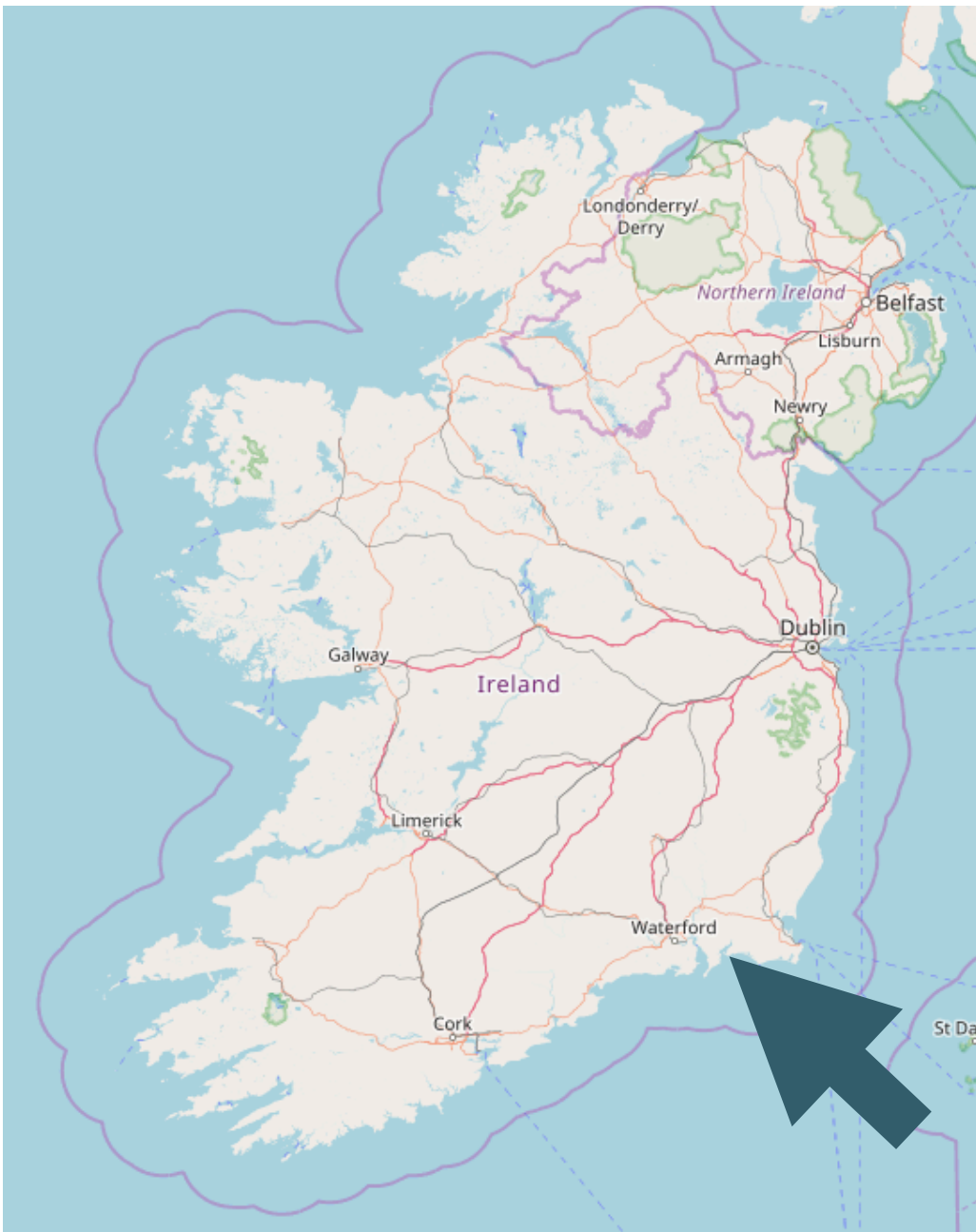


Eamonn de Leastar
edeleastar@wit.ie

<https://wit-oth-regensburg-2017-dmas.github.io>

<https://elearning.uni-regensburg.de/course/view.php?id=29928>

Waterford



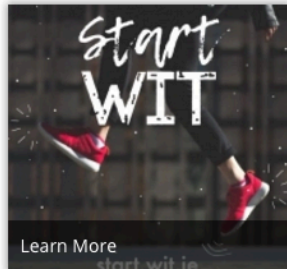
Waterford (from Old Norse *Veðrafjörðr*, meaning "ram (wether) fjord", Irish: *Port Láirge*) is a city in Ireland. It is in the South-East Region, Ireland and is part of the province of Munster. The city is situated at the head of Waterford Harbour. It is the oldest^{[2][3]} and the fifth most populous city in the Republic of Ireland. It is the eighth most populous city on the island of Ireland.



WIT A World of Imagination

IMAGINE WIT - VISIT OUR CAO HUB
Advice and information to help you research your CAO options

START WIT - REGISTRATION
& ORIENTATION



Learn More

PART TIME COURSES



Find out more

ACADEMIC TIME
SEMESTER 1

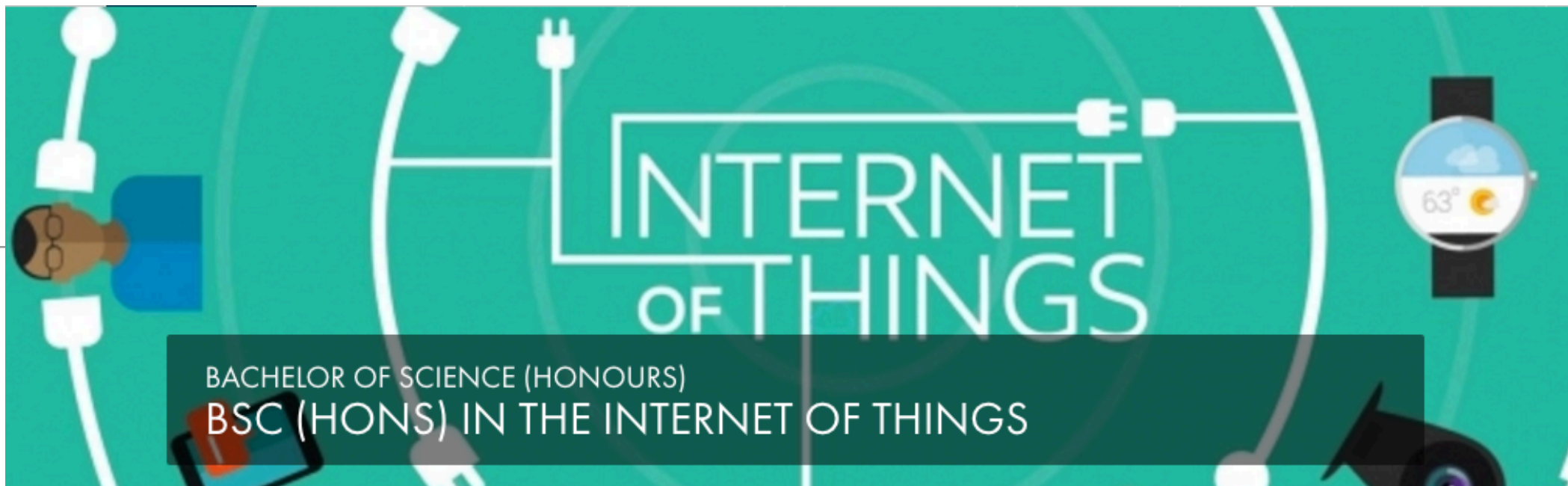


Click Here to View



Waterford Institute of Technology (WIT) is a university-level institution in the South-East of Ireland with over 10,000 students and 1,000 staff.

WIT offers tuition and research programmes in various areas from Higher Certificate to Degree to PhD.



[Home](#) » [Courses](#) » [School](#) » [Science](#) » [Department of Computing, Maths & Physics](#)

BY SCHOOL

[Business](#)

[Lifelong Learning & Education](#)

[Engineering](#)

[Health Sciences](#)

[Humanities](#)

[Science and Computing](#)

Code: WD197
Level: 8
Credits: 240 ECTS

Delivery: Full Time
Campus: Cork Road

2016 CAO range: 275 - 430
Duration: 4 years

APPLY FOR THIS COURSE



DESCRIPTION

[OUTLINE](#)

[ENTRY](#)

[OPPORTUNITIES](#)

[PROJECTS](#)

[STORIES](#)

[CONTACT](#)

Last Word Interview



Today FM Last Word Internet of Things Discussion

Matt Cooper, Eamonn de Leastar(WIT IoT Course Leader) & Barry Morris

0:00 / 11:44



- | | |
|--|-------|
| ▶ 1. Today FM Last Word Internet of Things Discussion by Matt Cooper, Eamonn de Leastar(WIT IoT Course Lead... | 11:44 |
| 2. Limericks Live 95fm: The Internet of Things by Shane McAllister, Eamonn de Leastar & Anthony Quigney | 20:07 |

Be part of the Internet of Things (IOT)

On this exciting new course developed in conjunction with the Institute's world class research groups TSSG, ACG, and CTRG, you will learn how to programme the next wave of connected devices, you will explore the software and hardware that is transforming the world, connecting things to create radical new services and products.

TSSG



ABOUT»

PROJECTS

RESEAR

[Home](#) » [About](#) » [People](#) » Eamonn de Leastar

Eamonn de Leastar , Board Member



Eamonn de Leastar

Board Member

Ph: +353 (0)51 302 965

Email: edeleastar@tssg.org

Personal Summary

Eamonn de Leastar is a co-founder and Chief Technical Officer of the Software Systems Group (TSSG).

After graduation in 1982, he taught computer science at Dundalk Ins he was responsible for a range of new curricula within this sector. He then moved into industry, gaining significant experience in senior programmer, analyst, quality assurance and project leadership roles. This included periods at Wordstar International, developing some of the first work processors on the then newly established PC standard and at Contel Business Systems



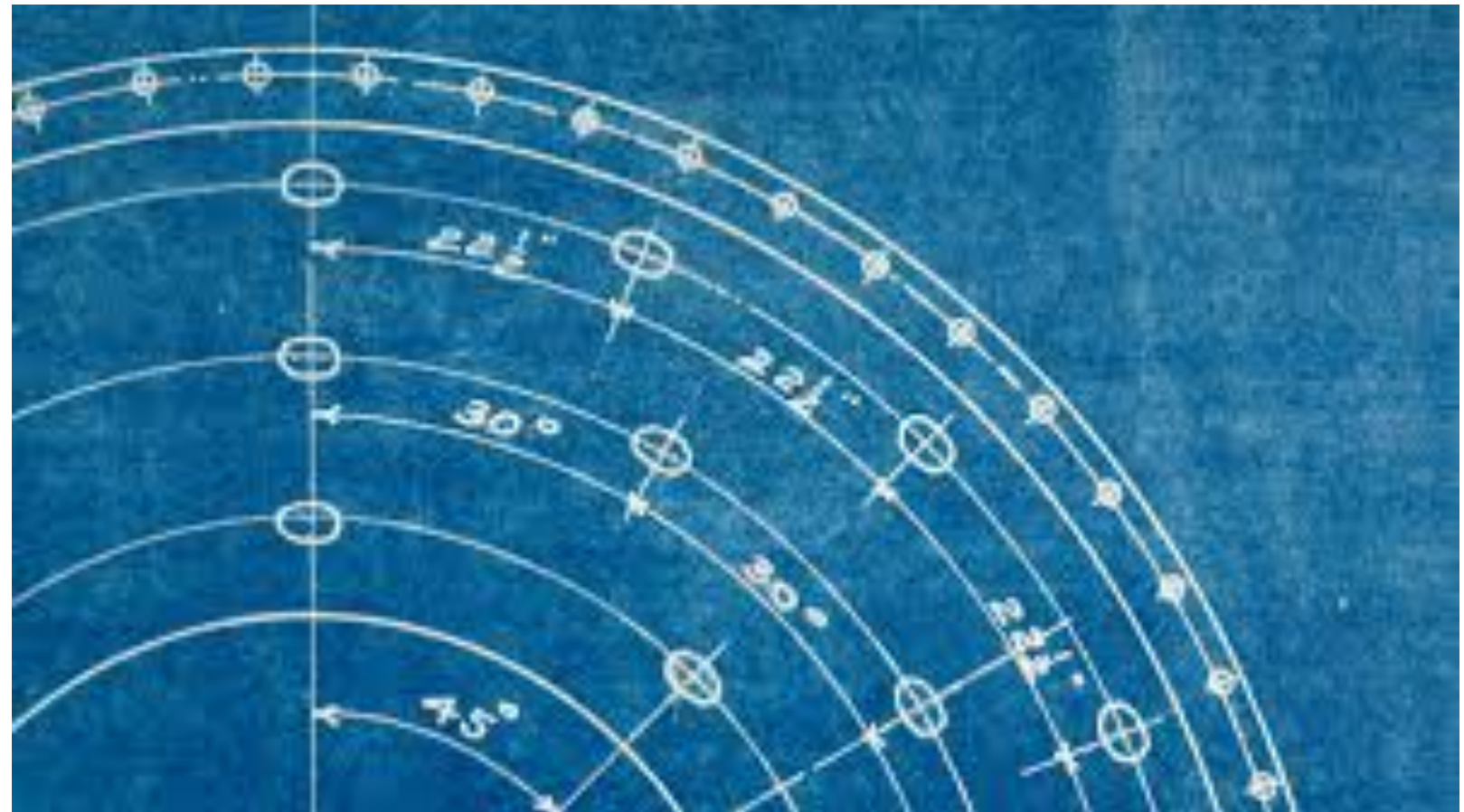
Course Mission

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

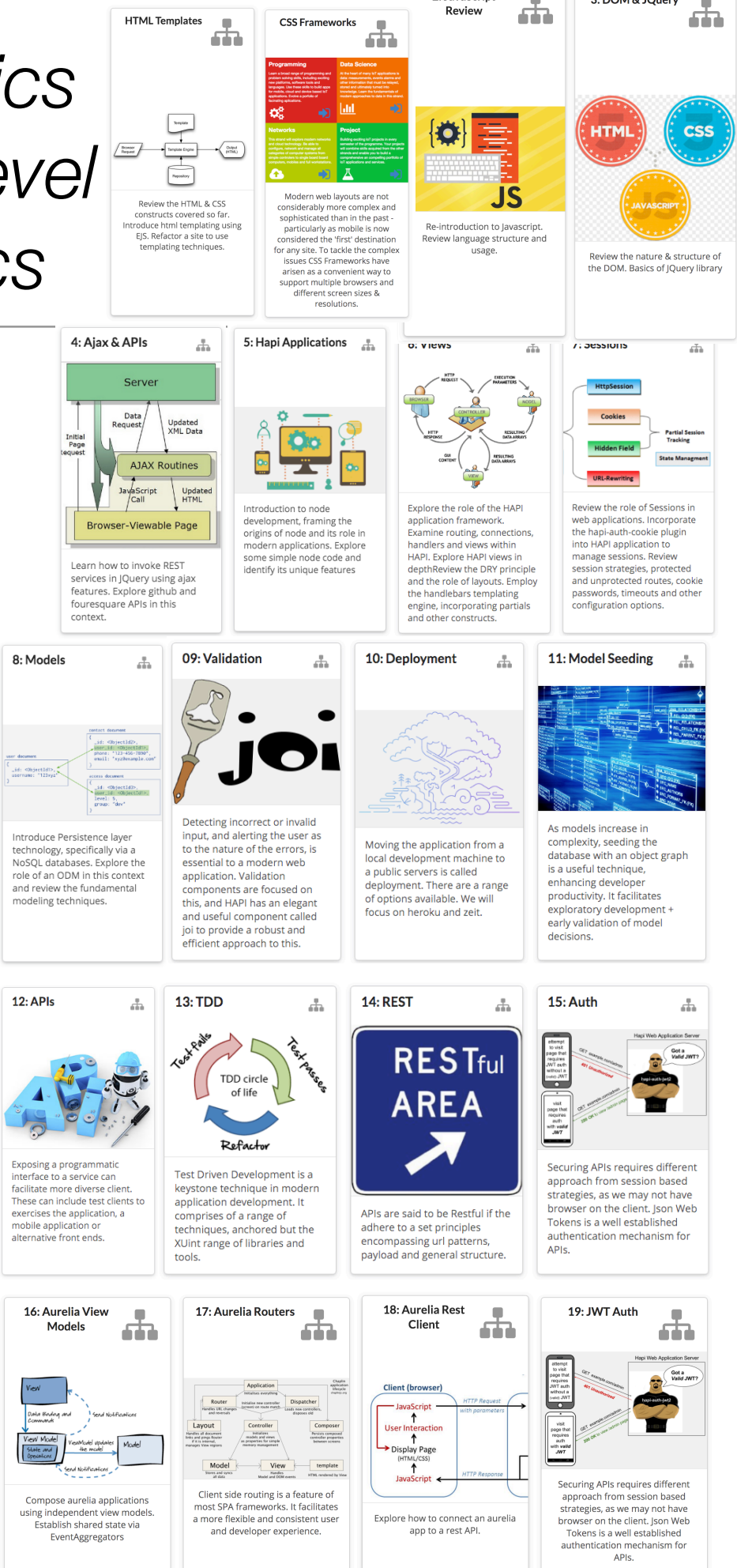
(1) Front End Foundation (+JS)

(2) Apis, Node & Hapi Applications

(3) Models, Persistence & Deployment

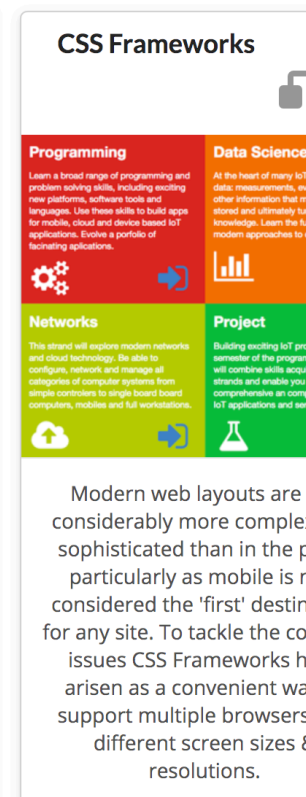
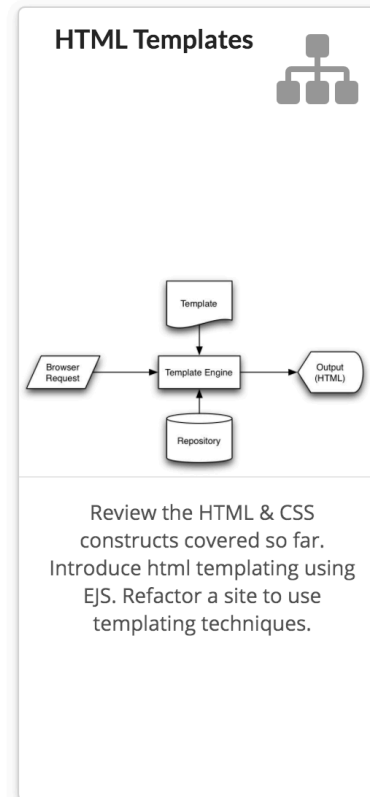
(4) Test Driven API Development

(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



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

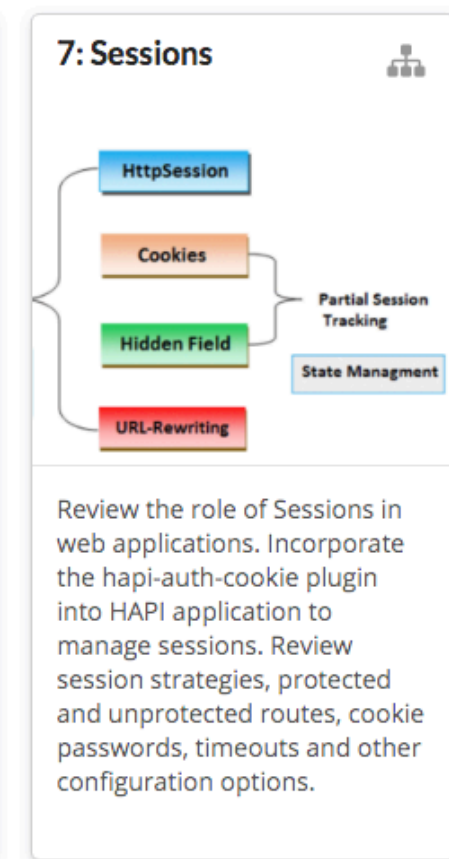
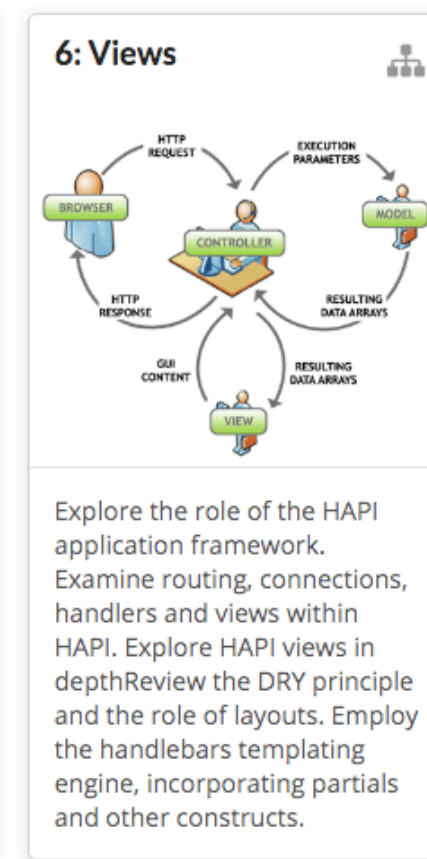
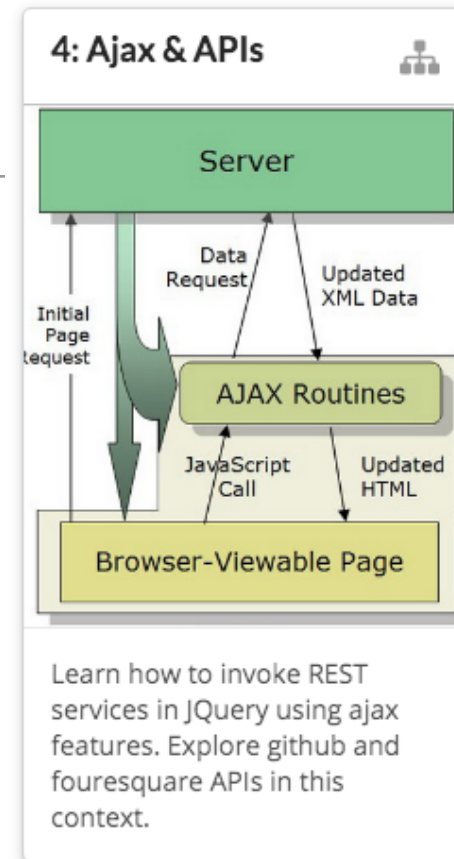
- Review Html + Css, focusing on templates + css frameworks
- Crash course in Javascript Fundamentals
- Learn basics of JQuery

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

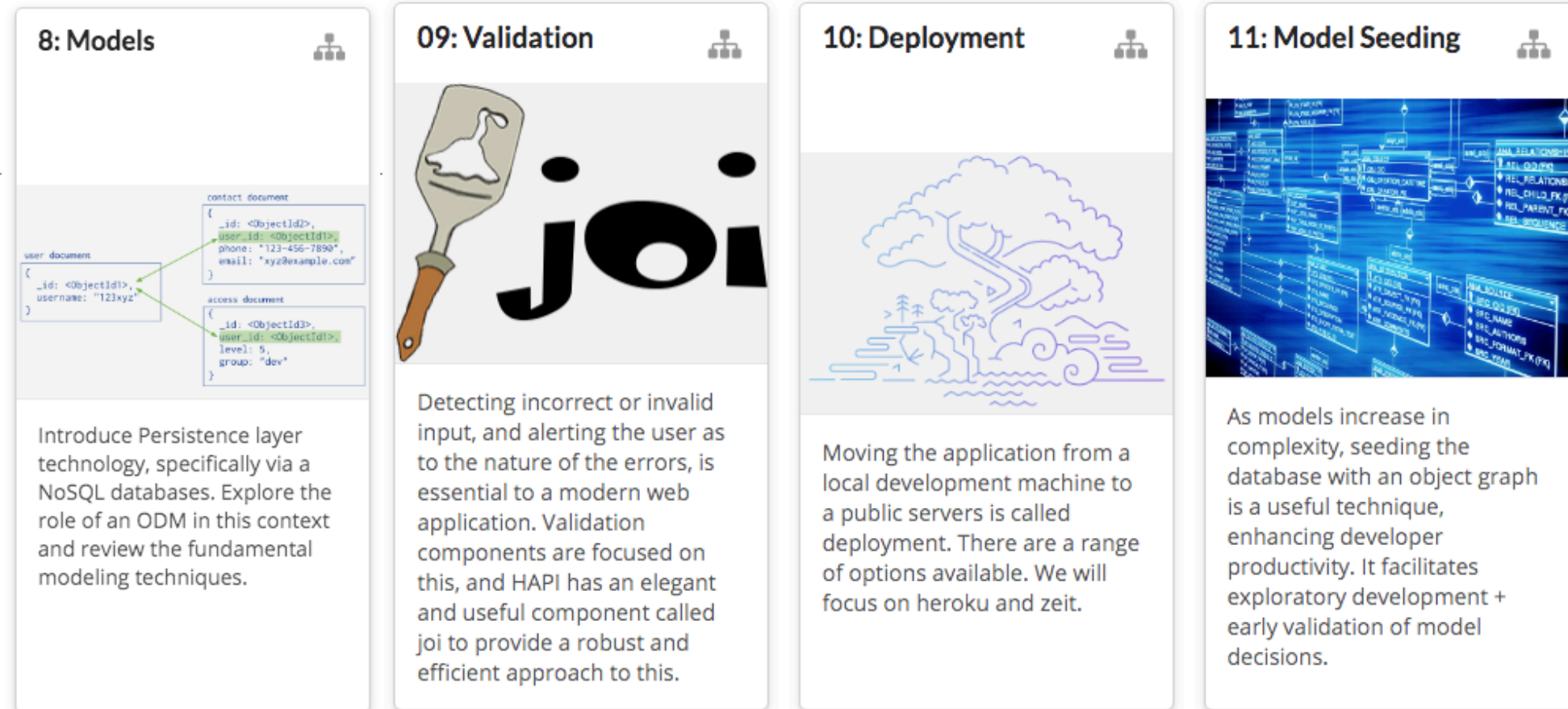
- Accessing APIs in Javascript
- Node.js Fundamentals
- Structure of a Hapi application
- Rendering views & templates
- Session Management

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



Concepts

- Models & Schema
- Database access
- Validation of data
- Deployment

Tools

- MongoDB
- Heroku toolbelt
- Mongoose, mongoose-seeder, joi

(4) Test Driven API Development

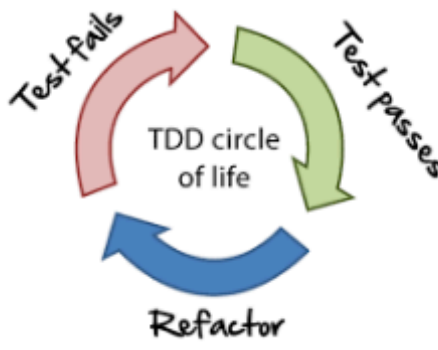
- Be able to design, implement, test and secure a Restful API

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.

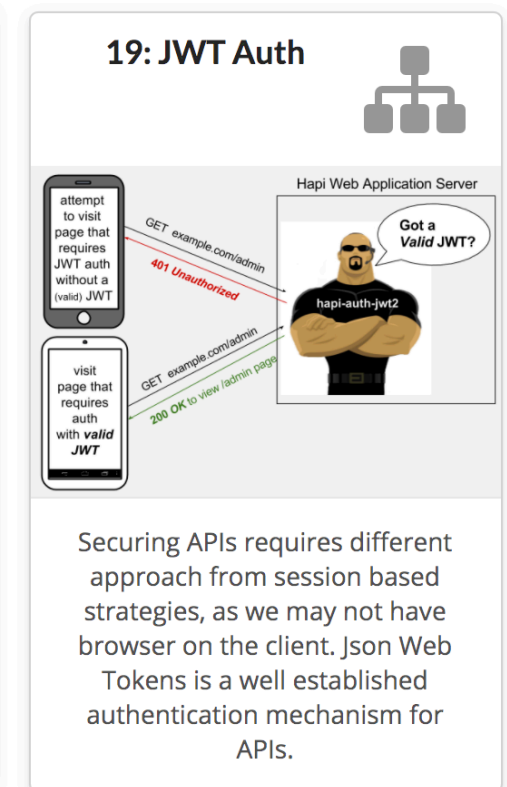
Concepts

- Constructing a basic API
- TDD, Theory & Practice
- Fundamentals of REST
- Securing REST APIs

Tools:

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

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



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

- 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 deliver rich UI experience.

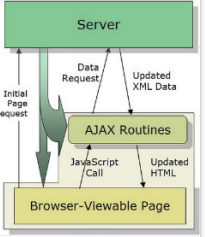
3: JavaScript & DOM



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

4: Ajax & APIs



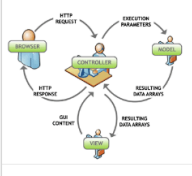
Learn how to invoke REST services in JQuery using ajax features. Explore github and foursquare APIs in this context.

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.

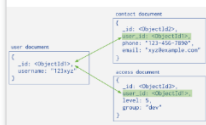
7: Sessions



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.

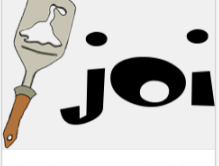
(2) Apis, Node & Hapi Applications

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.

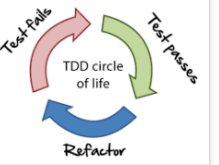
(3) Models, Persistence & Deployment

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 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.

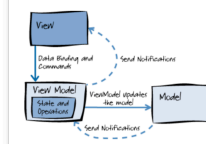
15: Auth



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.

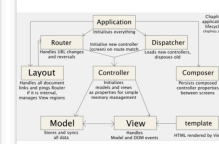
(4) Test Driven API Development

16: Aurelia View Models



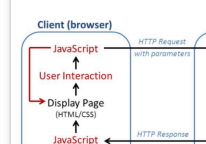
Compose aurelia applications using independent view models. Establish shared state via EventAggregators

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



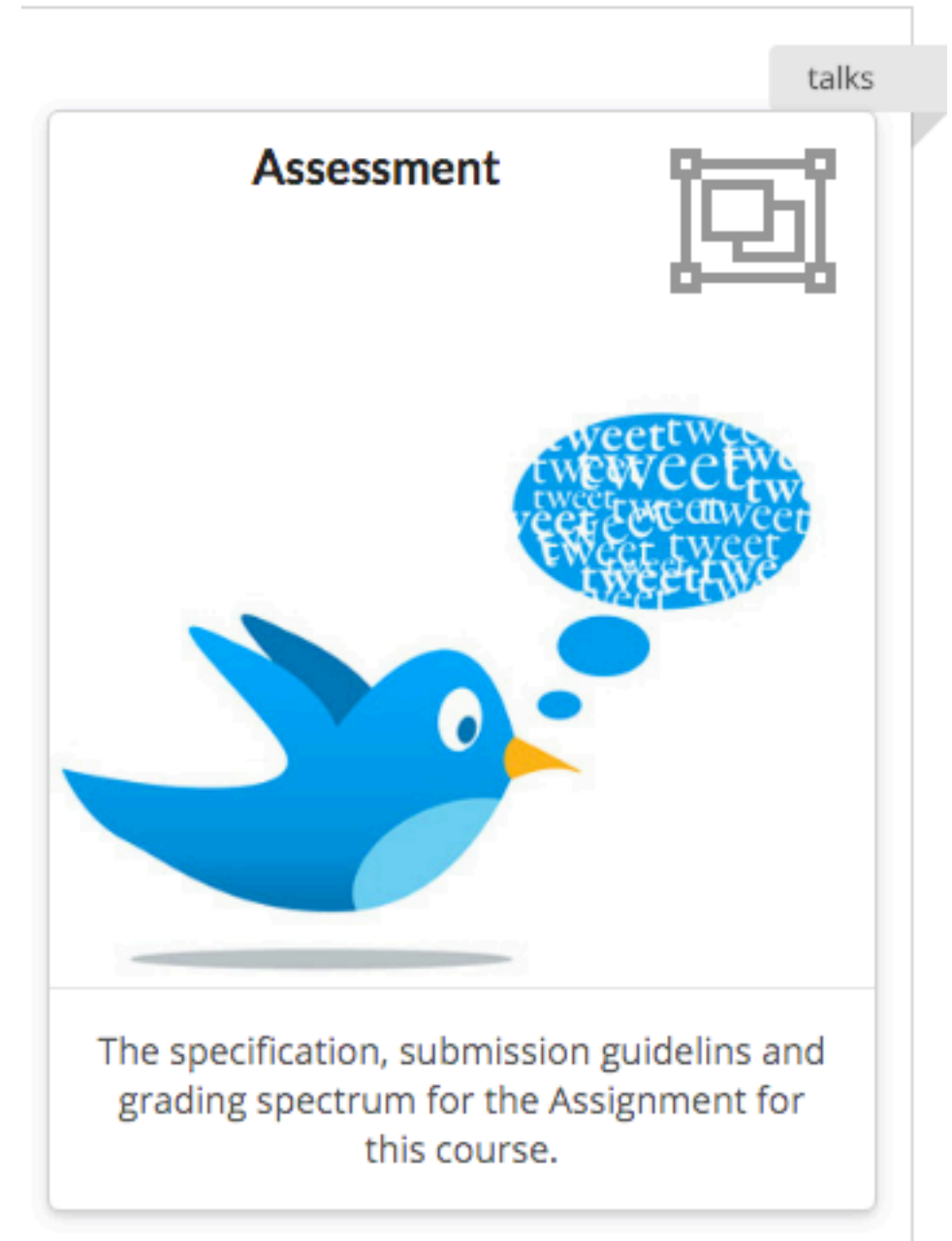
Securing APIs requires different approach from session based strategies, as we may not have browser on the client. Json Web Tokens is a well established authentication mechanism for APIs.

(5) Single Page Applications (optional)

Assessment

Single Project - submitted at
end of semester (late
December 2017)

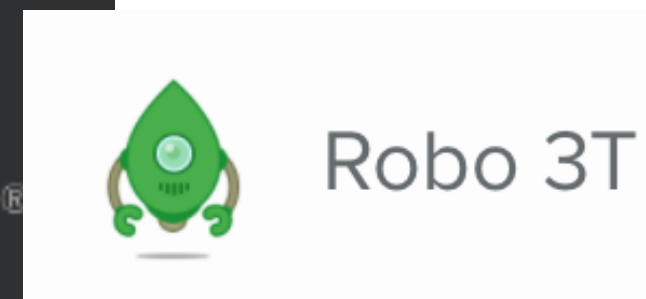
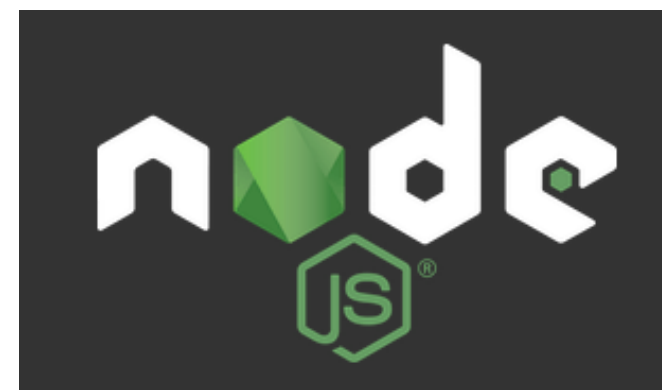
Assessment Specification



<https://wit-oth-regensburg-2017-dmas.github.io/topic-00-intro>

Lab Requirements

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



Schedule (pre and post tuition)

Pre Tuition				
	02/10/2017	09/10/2017	16/10/2017	23/10/2017
18:45 - 20:15	00: Course Overview	01 HTML Templates	02 CSS Frameworks	03 Javascript Review I

Post Tuition						
	06/11/2017	13/11/2017	20/11/2017	27/11/2017	04/12/2017	11/12/2017
18:45 - 20:15	18 Aurelia Intro	19 Aurelia View/Models	20 Aurelia Routers	21 Aurelia Rest Client	22 JWT Auth	23 Aurelia & JWT

	Sat:28h	
10:00-11:00	01-03 Introduction + Javascript Review	
11:00-11:15	Coffee Break	
11:15-12:30	04 Ajax + APIs I	
12:30-1:15	Lunch	
1:15-3:00	04 Ajax + APIs II	
3:00-3:15	Coffee Break	
3:15-4:45	05 Node + Hapi	

	On Site Tuition				
	Thursday 2nd		Friday: 3rd		Saturday 4h
10:00-11:00			08 Models		12 APIs
11:00-11:15			Coffee Break		Coffee Break
11:15-1:15			09 Validation		13 TDD
1:15-2:00					Lunch
2:00-3:15					14 REST
3:15-3:30			10 Deployment		Coffee Break
3:30-4:45					Assignment Briefing
4:45-5:00			Coffee Break		
5:00-6:15	05 Views		11 Model Seeding		
6:15-6:30	Coffee Break				
6:30-8:15	07 Sessions				

Discussion Forum also here:
<https://elearning.uni-regensburg.de/course/view.php?id=29928§ion=1>

[Home](#) ▶ [Courses](#) ▶ [OTH Regensburg](#) ▶ [Fakultät Informatik und Mathematik](#) ▶ [Internationale Gastdozierende](#) ▶ [YMWa-de Leastar-IN,IW-WiSe17/18](#)

QUICKLINKS

Meine Kurse (Dashboard)

Mitteilungen

Logout

NAVIGATION

Home

- Dashboard
- Site pages
- Current course
 - YMWa-de Leastar-IN,IW-WiSe17/18
 - Participants
 - Badges
 - General
 - 0: Overview & Assessment
 - Assignment
 - 1: HTML & CSS
 - 2: CSS Frameworks
 - 3: Javascript Introduction
 - JQuery & DOM
 - My courses

ADMINISTRATION

Switch role to...

Return to my normal role

MY COURSES

Course: Building Modern Web Applications & Services using Node.js – de Leastar – IN,IW – WiSe17/18, Topic: 0: Overview & Assessment

Nachrichtenforum

1: HTML & CSS ▶

0: Overview & Assessment

Your progress ?

Course Overview

1: HTML Applications

2: Views

3: Sessions

4: Models

5: Validation

6: Deployment

7: API's

8: Model Seeding

A brief overview of the entire course

Assessment



The specification, submission guidelines and grading spectrum for the Assignment for this course.

Assignment

20

<https://wit-oth-regensburg-2017-dmas.github.io>

Building Modern Web Applications & Services


Module Home

Prepared for OTH Regensburg by Eamonn de Leastar, WIT.
Creative Commons License

 17  9

1: Introduction

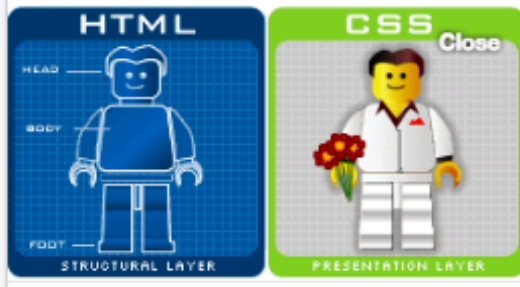




Introducing the module, assessment and schedule.

1: HTML & CSS Revision





Review of the fundamentals of HTML Templating + the Semantic UI CSS Framework

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

Ask Questions here:

OPEN CHAT

<https://wit-oth-regensburg-2017-dmas.github.io>

Public Chat Room:
Requires account/
sign in on *https://
gitter.im* first

I will drop in
occasionally to
answers any
questions or chat in
general about
course

Building Modern Web
Applications & Services

Module Home

1: Introduction

1. HTML: Templates
2. CSS Frameworks
3. JavaScript Orientation
4. Ajax & APIs

5. Front Applications
6. Views
7. Services
8. Models

9. Validation
10. Deployment
11. APIs
12. Model Testing

13. TDD
14. REST
15. Auth
16. SPA

Introducing the module,
assessment and schedule.

good day all!

Click here to type a chat message. Supports GitHub
flavoured markdown.

2016 course is still online.

However,
assessment and
content will be
refreshed

