Dec 2nd - 6th 2019 (Mon-Fri clas[s](https://meet.wellsfargo.com/linda.dixson/V2L1HHD9?sl=1))

**Introduction to Angular** at SSA in Columbia, MD

**Instructor**: Judy Lipinski

[judy@karmoxie.com](mailto:judy@karmoxie.com)

**Welcome to   
Introduction to Angular!**

To reach THIS document: (not .com and no www)

[**http://moxie.pro/2019dec2angular**](http://moxie.pro/2019dec2angular)

We will introduce the course and then get you connected to your lab machines.

**Contents:**

[Electronic Links to Chapters:](#_myzdap8wjzi1)

[Good resources for HTML, CSS and JS](#_queujmswuftt)

[Parking Lot:](#_8l3kowr88pr5)

[Music Requests:](#_b57dao5gv240)

[Tech Resources / Discussions:](#_q9hmgkvxhwfu)

[Misc:](#_6018c8lurbd4)

[Review](#_4az22vcys3mg)

We will introduce the course and then get you connected to your lab machines. From there you will be able to access this page.

# Electronic Links to Chapters:

# 

|  |  |
| --- | --- |
| 1 | [Overview of Course and Angular Development](https://docs.google.com/presentation/d/e/2PACX-1vRa09NPuxSQUOJS9e4bxPd9tTXcXMffYoU_by8gpCSwSAV-IH2J0Qsu2I2R97658UbR6yZSeWhJ6oM7/pub?start=false&loop=false&delayms=3000) |
| 2 | [Creating and Exploring Angular Projects](https://docs.google.com/presentation/d/e/2PACX-1vQcsbEnq0cJWoVfP7C5sZVpbtQ5OkKzSpA3mnh2GGQDK7mdqlnqXwsrUUON9-WShsUjBma8AeaSThCB/pub?start=false&loop=false&delayms=3000) |
| 3 | [Data Binding in Components](https://docs.google.com/presentation/d/e/2PACX-1vSxMHnb4n_mgttstyuYkvMdFZG98DHWBgBhpkblTikSTFLACkdMJqgGjOBZ_0oh1I2N_zpp3MeCd3e4/pub?start=false&loop=false&delayms=3000) |
| 4 | [Passing Data & Event Binding](https://docs.google.com/presentation/d/e/2PACX-1vSa47fAF0urZVCcz_HHKa60luV-yzguMYBji0lWOaDYeq-YWje0olCiyrdgjKBieN9SY1JWay9z10-s/pub?start=false&loop=false&delayms=3000) |
| 5 | [Using Directives](https://docs.google.com/presentation/d/e/2PACX-1vR055PAjjVep7vTlC30l-LTAk2KK65ylKW1uYLr3uZxzBr1320-6yNeg_d34GW_bTMZsKVFXSC3Ra93/pub?start=false&loop=false&delayms=3000) |
| 6 | [Using and Creating Pipes](https://docs.google.com/presentation/d/e/2PACX-1vQwNZE_oir_whJIwmMu-XUJMCbXazexBZZBRHpsr62NmkXLaqQiBpWZZI8KDrhVpFLpf_4tMaTPNNVR/pub?start=false&loop=false&delayms=3000) |
| 7 | [Understanding Change Detection and Lifecycle hooks](https://docs.google.com/presentation/d/e/2PACX-1vSgZZAuipN5mso_hhKRyquSzYkymv4QyIktK89Hpuln_zrtG1XR6bQtWEKSUF3f_4NWkqTPOTlK13hI/pub?start=false&loop=false&delayms=3000) |
| 8 | [Services, HttpClient and Observables](https://docs.google.com/presentation/d/e/2PACX-1vSDrFLTozxJS74CRuasJICuTf5pgyih9_HvrW5sqK6OStZaVcEuqp-G-HH1PcJ1sg-PJIpuT-x19O8E/pub?start=false&loop=false&delayms=3000) |
| 9 | [Leveraging Routing Features](https://docs.google.com/presentation/d/e/2PACX-1vQYAb4N1Phggpk5E--_5TZk3aqbE75HR1c6csvQ7N10NPBK7FOdJraYV1Jqz6fG5xwOmPvN6zyWKtkC/pub?start=false&loop=false&delayms=3000) |
| 10 | [Working with Forms](https://docs.google.com/presentation/d/e/2PACX-1vQnquKXsRPI2wRWpUYXkYJngbNS5et9q9XWNjY2Uaw5tIDLNBfECc3O9xefa-_FZZpewjFazgiVoPD1/pub?start=false&loop=false&delayms=3000) |
| 11 | [Unit Testing in Angular](https://docs.google.com/presentation/d/e/2PACX-1vQ77QDzfnYAB7BnEQGypeko8oCuvn24kPMSixby096xHLXrDa2C8YDmCu0_OeY0jGsJEd1_1pNPTqtR/pub?start=false&loop=false&delayms=3000) |
| 12 | [Summary](https://docs.google.com/presentation/d/e/2PACX-1vT1Ag7Z3geuqgXq4091t5cJPyTo1gl1mOGAZqU13rWSY6X7jO1mEvPdYWU4fbrpWf7mLYDJKFgx98kp/pub?start=false&loop=false&delayms=3000) |

**Connecting to the ProTech Training Virtual Machines**

1. [**Link to Protech VM connection presentation**](https://docs.google.com/presentation/d/1djTRHnVS0Etjrq6ScR3RRsVvOQPW-R1CUwi6YkgpCII/edit?usp=sharing)
   1. MAIN POINTS
      1. control+shift+tab for copy/paste
      2. only ONE connection allowed, no multiple tabs
2. Please use spreadsheet to find add your name machine assignment and check in:  
    [**Links to list of machine assignments**](https://docs.google.com/spreadsheets/d/147CPf86RmZFrJ8AxEZ-ofOICNAqaziwog0KaXU9zRr0/edit#gid=0)

Virtual Machine Login:

[**http://labs.protechtraining.com/**](http://labs.protechtraining.com/)

GitHub link to repos:

* git clone https://github.com/JudyLipinski/angular100-demos-2019-12-2
* git clone <https://github.com/JudyLipinski/angular100-labs-2019-12-2>
* git clone https://github.com/JudyLipinski/angular100-solutions-2019-12-2

# Good resources for HTML, CSS and JS

* W3chools: <http://w3schools.com>
* MDN - mozilla developer network <https://developer.mozilla.org/en-US/docs/Web>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

# Parking Lot:

* demo chrome debugger in vs code
* pausing Observables <https://medium.com/@kddsky/pauseable-observables-in-rxjs-58ce2b8c7dfd>
* <https://medium.com/js-in-action/rxjs-pause-and-resume-mighty-switchmap-41d0d1fe1113>
* 
* modal window for component

# Music Requests:

* study/focus <https://www.youtube.com/watch?v=WPni755-Krg>
* [Tommy Guerrero](https://www.youtube.com/watch?v=c7RM8TS9-Ps&list=PLUOcNXprlXYFet-Rmh3NKyLZsnaeFsM5z)
* Florence violin: <https://www.youtube.com/watch?v=7t3xBqAWLaU>
* St Germain <https://www.youtube.com/watch?v=PNp1PnfWQD0&list=RDEMd_6qMCbHagA2KqkA2AbbIA&start_radio=1>
* <https://youtu.be/w0RN8Hp8VXc> Japanese Trap & Bass Type Beats by GRILLABEATS ☯ 1 Hour Lofi Hip Hop Mix
* [https://www.youtube.com/watch?v=0RkSy6ElNIY](https://www.youtube.com/watch?v=0RkSy6ElNIY&fbclid=IwAR1X9j0jmDjj5dB4LWBmOgbxawc1defn0nMwLl1m-72CJCo4quM_fZNTCCU) carol of the bells

# [https://youtu.be/EKkzbbLYPuI](https://youtu.be/EKkzbbLYPuI?fbclid=IwAR1wZA2qIsWoL9eDGJ-J3m738HSq0JHMn7vj3sWE9neZlD1TUq91sRKn5r0) Deadmau5 - Ghosts n Stuff <https://www.youtube.com/watch?v=QV8eiSA4vqc&list=PLB190ED3675E75386&index=11&t=0s>

# <https://www.youtube.com/watch?v=NCSAYm8QrRQ&fbclid=IwAR1CEKNIPW5zEi995ukKuyo6QJjsl7_vh5YgeAFiGmNyr7McBkXDiT1wYEI>

* <https://www.youtube.com/watch?v=PXYeARRyDWk> Daft Punk - Human After All

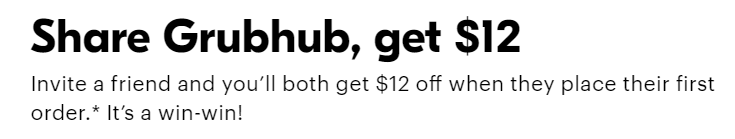
# <https://www.youtube.com/watch?v=lVKbF8khsrI> Daft Punk - Robot Rock / Oh Yeah

* <https://www.youtube.com/playlist?list=PLs3Svq6AjNCOKjzEcXNqrLAut312quZiL>
* Pachelbel Canon in D: <https://www.youtube.com/watch?v=hOA-2hl1Vbc>

# Tech Resources / Discussions:

* <https://www.typescriptlang.org/docs/handbook/basic-types.html>
* check out firefox extension for vs code debugging
* Emmet <https://docs.emmet.io/cheat-sheet/>
* Angular
  + <https://angular.io/guide/styleguide#lift>
* AngularJS to Angular Concepts: Quick Reference <https://angular.io/guide/ajs-quick-reference>
* Beware on VS Code - sometimes get error messages for Angular - that are not legit. see if your project still builds and works. if so shut down VS Code and restart - something must be wrong with some extensions
* Observables
  + <https://rxmarbles.com/>
  + <https://www.learnrxjs.io/operators/utility/do.html>

# Misc:

* <https://www.quora.com/How-is-Microsoft-Planning-to-make-money-from-GitHub>
* <https://insights.stackoverflow.com/survey/2019#key-results>
* ALMOST FREE LUNCH?

<https://www.grubhub.com/referral/efb6c10e-487e-11e4-9697-9cb654858910?utm_source=grubhub.com&utm_medium=content_owned&utm_campaign=growth_refer-a-friend_share-link&utm_content=promo_>

# Review

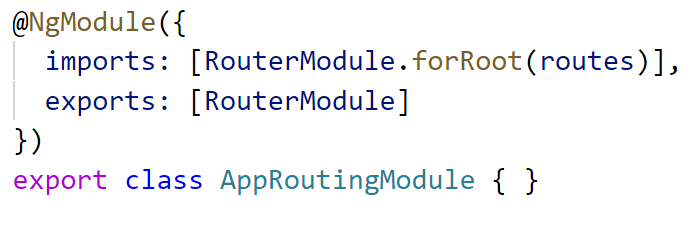
What is something new or that stood out for you yesterday? Any questions?

Fri

* json-server for creating a backend API
  + much faster for getting a REST API
* StyleGuidelines - LIFT
  + locate, identify, flat, Try to be DRY
* Routing
  + set up path objects Routes[ ]



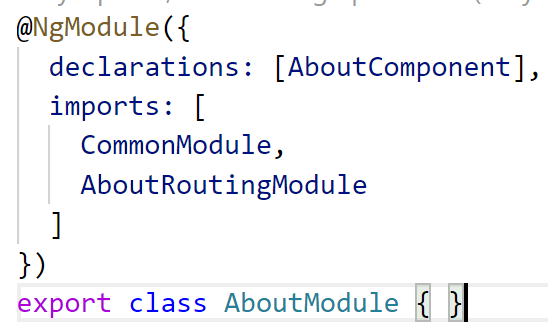
* + Need to add Routes to the RouterModule



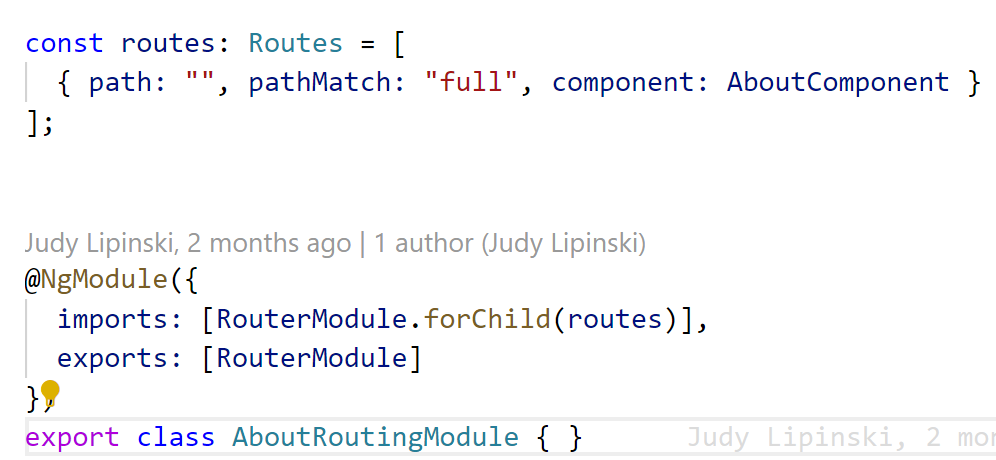
* + Add AppRoutingModule to project...



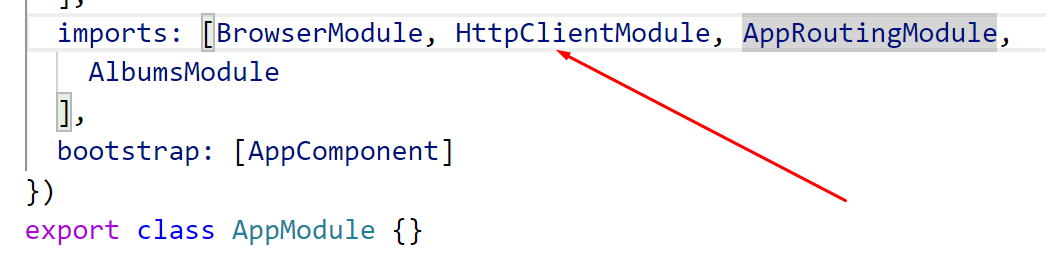
* + Lazy loading
    - AboutModule above is lazy loaded
    - where is component added project - not root but Feature Module



* + - where is the component to load specified?
      * use **forChild** in feature modules

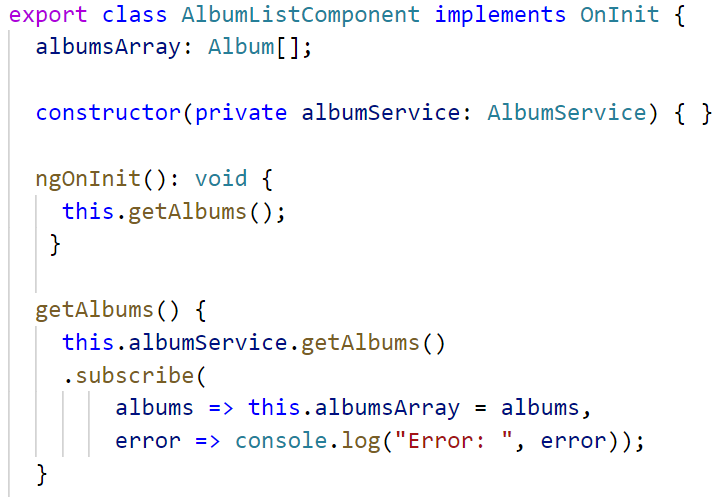


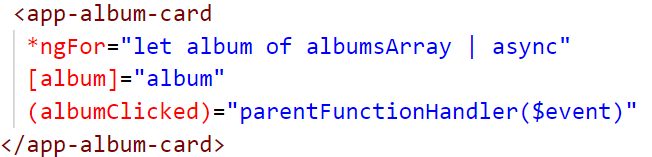
* Observables are LAZY
  + need to subscribe() for them to do anything
* RxJS - ES6 module for Observables and operators
* HttpClient is the module needed for AJAX
  + add HttpClientModule to root module

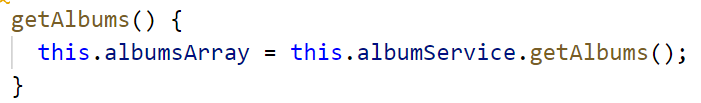


* In Service where you want AJAX call, inject HttpClient

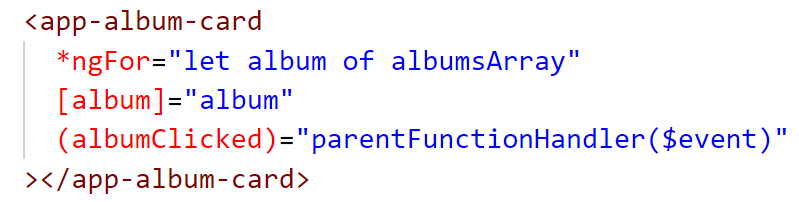


* + get() returns an Observable
  + providedIn: “root” - add Service to the Root module - used by the Root Injector
* In a component where you want to use it, inject a Service and call function
* 
* OR - instead of subscribing in component - can use async pipe - template use pipe and async



* in .ts file - don’t use subscribe - can use $ in variable name to make it stand out - ex. albumsArray$
* 

Thurs

* if you want to hide a component and not remove from DOM (uses CSS)
  + [hidden]=”booleanExpression”
* Directives
  + \* directives are structural - they manipulate the DOM
  + \*ngIf = “booleanExpression”
    - if true - add to the DOM
    - if false, not in the DOM
      * if it is being removed, the lifecycle method ngOnDestroy() is called
  + \*ngFor - allows us to iterate over an iterable/array
    - repeat the host element - for each item in array
    - 
  + Styling directives
  + [class.some-class]=”booleanAttribute”
  + [style.background]=”booleanExpression ? red: blue”
  + ngClass = “objectWithClasses”
    - objectWithClasses = {

class1: ‘booleanExpression1’,

class2: ‘booleanExpression2’,

class3: ‘booleanExpression3’,

}

* + ngStyle
* Pipes
  + Let us transform the data for display
  + doesnt change the data itself
  + {{ price | currency }}
  + {{ price | currency : ‘EUR’}}
  + {{ hireDate | date : ‘mm/dd/yyyy’ }}
  + ng g p name-of-pipe
    - creates a .ts file with one function
      * called transform(value)
        + return transformed value
    - @Pipe({name: ‘pipe-name-to-use’})
    - the Pipe gets added to the project where?
      * @NgModule ({declarations: [components, Pipes, custom-directives]}) //things used in HTML
* Services
  + a place to store reusable code
  + @Injectable is the decorator for a service
  + Services registered with root are Singletons
  + components get access to this singleton by being Dependency injected via the constructor
  + can use Services to share data between components
* Event Binding - uses <button (click) = “doStuff()”>
* Pass data from parent to child
  + @Input() in the child
    - @Input() **name**;
  + in parent: <app-child [**name**]=”passedName”>
* Event emitting from child to parent
  + @Output() in child
    - @Output() **hobbySelected**: EventEmitter<dataType> = new EventEmitter<dataType>
    - to raise the event:
      * this.**hobbySelected.emit()**
  + in parent: <app-child (**hobbySelected**)=”parentFunction()”>
* Observables
  + uses publish/**subscribe** model from OOD
  + can be set up to return just once, or multiple values over time
  + LAZY!!!! until something subscribes, nothing happens
  + Used throughout Angular APIs - HttpClient
  + asynchronous processing
    - AJAX - HttpClient
    - Form interaction
  + different than Promises
    - must subscribe
    - keep receiving data over time
    - can cancel observables

Wed

* CLI tool Add more components
  + Ng g c name
    - Name.component.ts
    - Name.component.html
    - Name.component.css
    - Name.component.spec
    - App.module.ts gets modified
      * Component is added to @NgModule({declarations: [Component]})
  + Can add new modules, services, pipes
* Create a hierarchy of components
  + Via the template (HTML)
  + <app-name>
* Changes to array or object property will not cause Angular to update the screen
  + Use the spread operator
* Use of interfaces
  + Type checking
  + If want optional, can use ?
  + Can indicate a property must be of a certain type or types

export interface person {

name: string;

age: 20|21|22;

isTrue: boolean | Promise<boolean> | Observable<boolean>

}

* Augury plugin for chrome and firefox
* Can restrict to a set of values
* Use of arrow functions
  + especially helpful if you are in async code - and want to use the keyword **this**
* ngOnInit()
  + where we initialize our components
  + this.property = GET DATA
* change detection cycles are triggered, with:
  + timer functions (setInterval, setTimeout)
  + browser events (click, mouseover)
  + AJAX
* 2 ways to do data binding
  + [ html-attribute]=”class-property” - can be a member variable or function call
    - [src]=” ’hard-coded literal value ’ ”
  + {{ interpolation }}

Tues

* Git Bash does not prompt when using ng new for the routing or CSS
* using package.json scripts
  + we added flags to ng serve to set the port and -o - opens browser when web pack server launches
  + changed favicon to distinguish projects in browser tabs
  + CHANGED PORT:
    - to have more than one ng app running
    - favicon is cached by browser by port #
* review of modern web dev course was helpful for those how havent had modern web dev in a while
* VS Code
  + GitLens extension
  + Source Control within VS Code
    - committing code
  + Peacock for coloring windows
  + Font icons
  + Emmet abbreviations
* Angular building - production versus development mode
  + this has been a focus for ng team last few major releases
* Angular
  + Angular CLI
    - does so much setup
  + Decorators
    - @Component
* TypeScript
* stackblitz - can demo Angular code