ng serve to open app in browser localhost:4200 in TODO folder
(Control+C to stop)

ng lint - checks code for coding standards tslint.json has all rules listed.

ng build - creates dist folder - can put this behind any server and run app.
Ng dist - put on server to deploy the app. Copy this file only to server

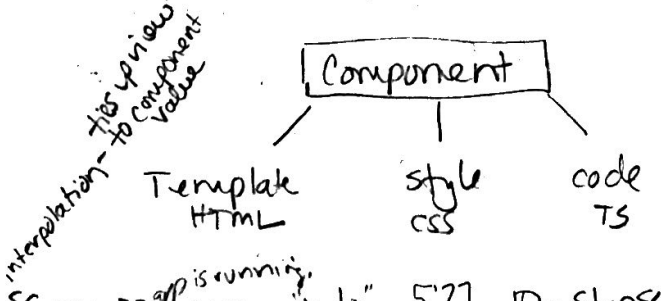
ng test - runs tests for angular in framework for Jasmine karma
test = specs karma.conf.js etc). spec.ts. Tests Component

ng e2e - end to end test - launches + tests entire app. Selenium
test usual for all languages. Protractor

Raleigh, NC.
Greensboro, NC.

App - All angular code
Assets - all pics
Environments -
tsconfig.json - converts typescript so js can be read.

Home Page
Login Page
Welcome Page
Todos Page



Data Binding {{title}} app is running, title = "todo" 5:27 ID - Step8

ng generate component welcome - can't roll this back

- Every Angular component (@Component) has to be associated with an Angular Module (@NgModule)

- TypeScript - Interface OnInit - describes method ngOnInit
- decorator (annotation in Java) @Component

Control / to comment out hilighted code.

Data Binding + Event Binding - 2 way data binding

When referring to member variables you MUST use this.

Step 14 - Banana in a box [(()] ng Model - Part of forms module

★ Step 17 - how to add multiple page routes app-routing.module.ts
3:00 Const routes array - add another path. Most have added the components already, import them into app-routing. Don't forget comma on array!
- must put tag in app.component.html to activate routing
- added error component.

Step 18 Dependency Injection - implement navig. from login page to welcome page
(constructor)

Step 19 Add name Route Param for Welcome Component
Add username as a param: so it will show up on login page
2:25 this.route.snapshot.params['name'] - params is a Map, name is the Key

Step 20 - Create List Todos Component w/ ng generate
Order of routes in app.routes is very important. All must be above the ** route
Add array todo list in list todos.components and then MAP to todos.comp.html
Use interpolation {{}} to link todo.id, todo.description. Loop through array using * ngFor - declaring a variable todo of the type todos (7:00

8/14 - 10:50pm
Step 21 - Create a Link to Todos in Welcome Component

Step 22 - Best Practice - Create a Todo Class - Pipe to format the date. Convert one format in a field to another.

Step 23 - Intro to Angular Modules - Comparison to Javascript

Step 24 - Understanding Bootstrapping of Angular App w/ Root Module + Compon
1st Module that is bootstrapped is the root module (main.ts) app-root index.html
app.component.html displays on all pages. Set header + footer here.

Step 25 - Quick Rv. Angular Modules + Components Tasks to Complete.
1. Nav + menu bar
2. Formatting
3. Security for Menus
4. No hard coded logic in Todolist + Components
5. Functionality - Edit Delete Add
6. Spring Boot
7. Spring Security

Step 26 Overview of Next Steps - BootStrap Menu, Footer, Refactoring
added bootstrap link to style.css
ng generate component menu

step 27 Adding bootstrap framework + Creating Menu + Footer Component

Step 28 Using bootstrap to Create a Menu w/ Navigation Links

Step 29 Styling Footer + other components w/ CSS and Bootstrap

---

8/16/2020

Step 30 - Good Practice - Use routerLink instead of href for Routes - using routerLink allows page not to be totally reloaded when clicked on like href. Single Pag application works best this way. Have to use href to go to external website

Step 31 - Creating an Independent Authentication Service Component
"ng generate service" to create service component. Substitute for Auth svs w/ JW authenticationetc. Deleted... only to show how to add them. To create the svs folder + put files in them the command is
ng generate service service/hardcodedAuthentication
@Injectable makes file a service. Can inject hardcodedAuth class anywhere we want.

Step 32 - Using Session Storage to Store User Authentication Token - creating a way to store that the User has logged in. Session Storage. Associated w/ a browser session

Step 33 - Enabling Menu Links Based on User Authentication TOKen

Step 34 - Implementing Logout to remove User Authentication TOKen
ng generate component logout - created logout method

Step 35 - Securing Components using Route Guards Part 1
App routing module is where we define routes. Create RouteGuard Servio
ng generate service service/routeGuard. Must implement Can Activate interface in its file to work

Step 36 - Part 2

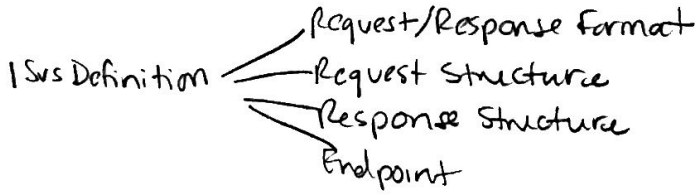
Step 37 - Review

Step 41: What's a Web Service

Step 42: Important How Questions related to Web Services

Request + Response: XML - extensible markup Language
JSON - Javascript Object Notation
↳ must be platform independent.

All web services follow this chart:



43. Web Services - Key Terminology

44. Intro to RESTFUL Web Services

@ angular/common/http

REST - Representational State Transfer
HTTP = Hyper Text Transfer Protocol
URI - Uniform Resource Identifier

45. Initializing a Restful Services Project w/ Spring Boot
Spring 2.3.1
Dependencies- Spring Web
Spring Boot DevTools
Spring Data JPA (SQL)
H2 Database (SQL)

8-17-2020

Step 46. Creating a Hello World Service
Cntl + n to open up menu to start a new class

Step 47 Enhancing the Hello World Service to return a Bean
Create a hello world object & try to return back. Getters + Setters

Step 48 - Quicker of Spring Boot Auto Configuration + Dispatcher Servlet. Debug Log
Front controller URL - get request

Step 49 - Enhancing the Hello World Service w/ a Path Variable
id is a path parameter. Path vars are impt to REST

8-22-20

Step 50 Connecting Angular FE to SB REST SVS 1 - Creating Data Serv
HTTP Client Module - Framework - Observable - Welcome Screen w/ button
Call Data Service - Back-end API
ng generate service service/data/welcome Data imported to welcom component.ts
Executed hello bean world service

Step 51 step2 - http Client Module
HTTP Client - invokes get post delete put - HTTP Requests. Add to constructor
import HttpClient into app module

Step 51 - Observable recorded when button clicked No communication b/n server though
cont'd

Step 52 - Step3- Understanding Observables
Must call http svs asynchronously - Angular makes observables as interface for asynchronous operations
Hello Bean World Svs not executable till do a subscribe
- had to do annotation @ CrossOrigin(origins="http://localhost:4200") so SpringBoot would allow communication from 2 local host servers.

Step 53 - Step4 Understanding Subscribe
handleSuccessful Response function ng*If welcome.html interpolation

Step 54 - Step 5 handling Error Response
use Subscribe message to throw error in welcome.comp.ts
throw new exception in world bean controller
handle error Response in welcome component.ts

Step 55 - Calling Welcome HTTP Service w/ Path Variables
tick ${name} to declare in 28 min as variable path
invoke external svs

Step 56 - Designing Restful Svs for Todo Resource
Task List
1. Retrieve all Todos for a User = GET (method) /users/{user_name}/todos
2. Delete a Todo of a User = DELETE/users/{user_name}/todos/{todo_id}
3. Edit (update) a Todo of a User = PUT /users/{user_name}/todos/{todo_id}
4. Create a new Todo of a User = POST /users/{user_name}/todos/

Step 57 - Creating REST API for retrieving todo list
Todo Resource
Todo.java - id, username, descr., targetDate, isDone + getters + setters + constructor
Hard coded svs Class to act as database
Create @Service to manage instances through Spring

Step 58 - Connecting Angular Frontend with Todo List Restful Svs
Create Data Service
ng generate service service/data/todoData
add to list Todos component
Add cross origin to Todo Resource

Step 59 - Creating REST API to delete a Todo 1 Create Delete Request Method
Delete method added to TodoResource + Hard coded Service
generate hashcode + equals methods in Todo.Java
ResponseEntity helps build request w/ specific status

Step59 - Creating Rest API to delete a Todo 2 Execute Delete Req. method
Talent API - formerly Restlet
204 - no content

Step60 Adding Delete Todo Feature Angular Front end
Add delete buttons to list-todos.component.html
Add delete todo method in list-todos.ts
Add delete todo in todo-data-service
Add success msg to list todos component.html

Login
Home
Welcome
Todos
Update Todos

Step61 Creating Todo Component and Handle Routing
Create new screen to go to to update todos
ng generate component todo (to edit + create todos)
route this Component to todo update page

Step 62 Designing Todo Page w/ Bootstrap Framework
Create form to enter description + date todo.component.html
Save button. Invoke Click method saveTodo
SaveTodo broke code, can't compile

Step63- Creating retrieve Todo Service and Connect Angular Frontend
updated todo resource.java
updated todo-data.sv w/ retrieveTodo saveTodo still not working

Step64- Improve Todo Page Appearance
error - says retrieveTodo is not function
Found ERROR! Missing ; in todo-data.service.ts! on retrieveTodo & deleteTodo!
event binding for date
[ngModel] property binding
(ngModel) event binding pipe date format

Step 65 - Creating Rest API for Updating Todo - PUT Request Method
Created Save method in TodoHardcodedSvs.java
Create methods in Todo Resource. - one add, one update
(TodoUpdated compile error)

Step 66 - Creating REST API for Creating a Todo - POST Request Method
① Tested Get Request w/ Talend API Tester
② add default constructor in Todo.java
③ Check Post method [ServletUriComponentsBuilder]

Step 66 - Restful Web Svs Best Practices

Step 67 Implementing Update Todo Feature in Angular Front End
Save Todo finally works... had to create the method!

Step 68 Implementing New Todo Feature in Angular Frontend
Noticed no dropdown menu for all list on todo-list page

Step 69 Improving todo Form - Validation + form Submit on Enter - ng submit
add ngSubmit to todo component.ts + moved saveTodo up to ng Submit - removed click
template variable to check that form is valid and check logic around it. #todoForm
min. character towards end

Step 70 Enhancing Validation Messages on Todo Page
todo.component.css red line on form when invalid
decide what validation messages you want to show

Section 7
Step 71 Overview of Security w/ Basic Auth & JWT
Securing the app we have built. Start w/ Basic Authentication - send
user Id and password. Temp token - Json Web Token. JWT is
more secure.

Step 72 Setting up Spring Security
Add spring boot starter security dependency
Now on localhost 8080 it asks for un + pw log in when type in usr URL
Form based authentication
Default UN = user
PW is listed in console) Access Denied - Exception couldn't find either
Found issue! In application.properties needed to change -debug to = info

Step 73 - Configure standard userid and password
Tdd to install Spring Tool Suite 4 - installed using Marketplace!

Step 74 Enhancing Angular Welcome DataSvs to Use BasicAuth. sendverification headers

Cite issue

LeClair
Lakshmi
Star
Lionel

2.6% int'l rate
30 yrs.
ChromeOS
79

Step 75 - Configure Spring Security to disable CSRF and enable OPTION Requests

- Fix denial(401) of options request, CSRF Cross Site Request Forgery
- opened web security Configurer Adaptor. Class
- showed the configure(HttpSecurity)method.
- Copy ↑
  disable csrf token, we'll use jwt token to prevent cross side forgery requests
  allowpre-flight request. /** means any URL, antMatchers
  spring security configuration

Sept Rd Mtg 4 Thur Sept. Socrates forms

Ques #33
javax.persistence.EntityManager
Don't generate system times using code. Violation of how spring is supposed to work.

Step 76 - Creating Angular Http Interceptor to add Basic Auth Header
working@start
Http intercept to manage headers better - add header to every request
error 404 at 8080

Step 77 Configure Http Interceptor as Provider in App Module

Step 78 Create Basic Auth Restful Service in Spring Boot

Step 79 Create Angular Basic Authentication Service
had to enable experimental Decorators

Step 80 81 Connect Login Page to Basic Auth Svs Part 1/2
observable - asynchronous call
.pipe
imported map
Debugged - needed the Main App to point to the correct Controller (I have HelloWorld AND BasicAuth. Had to add @ComponentScan(basePackages={"packagey controller"}). Change later to Basic Auth

Step 82 Refactoring Angular Basic Auth Svs

Step 83 Refactoring HttpInterceptor to use Basic Authentication Token

84 Constants for URL CORS=Cross Origin Resource Sharing

Basic Auth Controller works w/ frontend page to work, still can't get all pages to work!
Hello World Controller allows backend login page to work

Case Study

Step 88 - ① I disabled @ComponentScan: let all controllers run. I did the get req. using Talend to /user/in28minutes/todos and did see the 401 msg that a JWT token is needed to access.

② Send post request to /authenticate add the following to Body

```
{
"username" : "in28minutes",
"password" : "dummy"
}
```

③ Got response back w the token. put in JWT + mySecret is token
Authorization, Bearer then token to users/in28minutes/todos, get request - successful!
w the login we ask for a token + use for request. Talked about how to refresh the token

Step 89 - Understanding JWT Spring Security Framework setup
What's left to do:
BCrypt Password Encoder - encode pwt store it
Statelessness - no sessions - REST API
Authentication Entry Point - no unauthorized users
JWT Token Authorization Once Per Request Filter
/h2-console - memory database
We are extending JWTWebSecurityConfig

Step 90 - Creating a new user w encoded password
Created Bcrypt class + imported Bcrypt pw = password@!23C#!
Posted new UN + pw in Talend and succeeded w token

Step 91 - Using JWT Token in Angular Front End
http-interceptor - basic-auth.service.ts
Ln27 Authorization: misspelled! :/

Step 92 Database name + URL from Console log - database name randomly generated
Make database URL a constant

FreeCam to record App

Step 92 Setting up Todo Entity + Populating Data

Import @Entity & @Id into todo.java

Wrapper class, not primitive long to Long

Changed counter to long in todo hard coded svs

@GeneratedValue to generate Id #

app.properties - sql + h2console

8080/h2-console | JDBC URL jdbc:h2:~/test - change to

jdbc:h2:mem:testdb - database is visible!

Added data through sql file and populated just fine

Step 93 Connecting Get Rest APIs to JPA Repository

TodoJpaResource

TodoJpa Repository

Step 94 - Connecting POST, PUT, and DELETE APIs to JPA Repository

objects ===

primitives == best

New - Create Register Page

1. ng generate component register
2. app-routing.module.ts
   a) add path: 'register', component: RegisterComponent}, added import

   register works shows up.
3. Add div on Login page w message to register.