קורס אנגולר 8 – CORE NET

מרצה : שמעון - - sdahan@jbh.co.il

05446490218

Core מחליף את ה-FRAMEWORK !

ב-JS אין משתנים ברמת IF רק ברמת פונקציה

Let החליף את VAR וכך יש משתנים ברמות שונות (מגירסה 6)

TS תומך בכל הגירסות של .JS

יש קובץ CONFIG ל-TS מגדירים שם לאיזה גירסה של JS אתה רוצה לתמוך.

TSC – ממיר מTS ל-JS

TS אפשר לכתוב JS רגיל אבל להוסיף מחסומים שיעצרו אותנו בשלב **הקומפילציה.**

למשל :

Function sum(arr:array<number>) :number

פונקציה שמקבלת מערך של רק מספרים ומחזירה רק מספר.

יש ב-TS O.O , אינטרפייס , למדה וכו'

Angular

מה מקבלים ?

Navigation,,Forms,Validation ,http,events,components,directives

עבודה במודולים

מודול מורכב:

component

, directivs

, pipe

**Component –** יכול לייצר תווית HTML חדשה ( כותרת SELECTOR ) שיכילו :

1.html

2.TS

3.CSS

(כמו כן יהיה אפשרות input וoutput)

Scss מתקמפל ל-CSS.

Directive – att חדש שאפשר לכתוב ב-HTML.

למשל תיבת input שאפשר לכתוב רק בעברית בה.

NPM - ספריות ל-JS

Tslint - תוספת ב-VSCODE שעוזרת שלא נעשה טעויות בקימפול , מונעת תקלות (מציק)

קובץ ב-TS , נקרא TS מודול. אם רוצים לחשוף אותו צריך להוסיף לו EXPORT. שזה שם של קובץ בעצם.

ואז בדף אחר שאנחנו רוצים לקרוא לו אז נשתמש בimport filename

מה הקשר בין TS לבין ה-HTML?

[] () {}

{{}} = innerhtml

[] = השמה לATTRIBUTE של HTML.

() - EVENTS []

הרעיון הוא לשנות את המראה ב-HTML והלוגיקה התיכנותית תהיה רק בTS

יש גם \*NGFOR הכוכבית אומרת שיש פה שינוי מהותי בDOM

תרגיל ראשון גלריית תמונות

תוספים לVSCODE

-ANGUAR LANGUAGE SERVICE- נותן אובייקטים בHTML

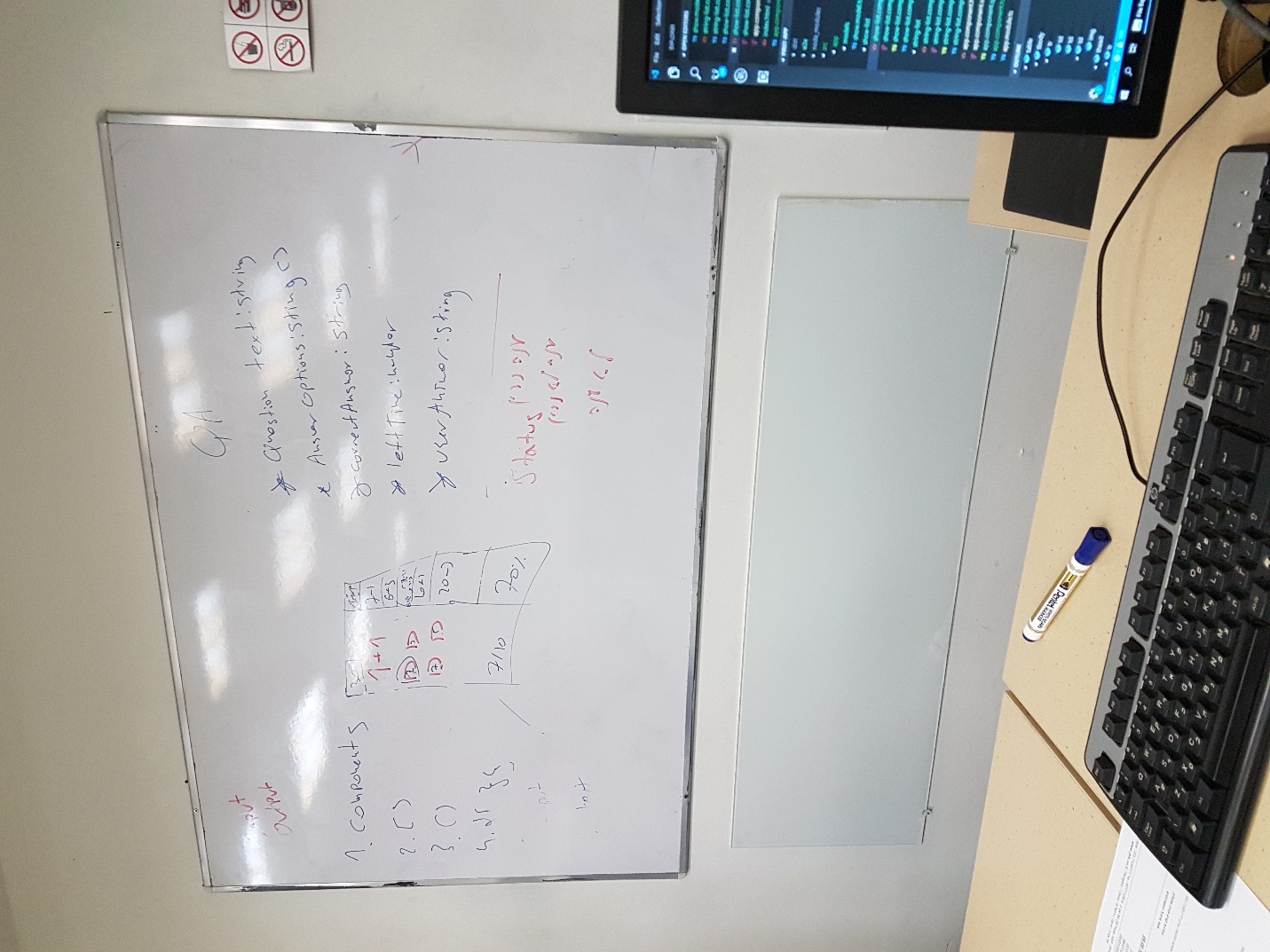
ANGULARS-SWITCHER – מחליף בין TS ל-HTML

Ng g n מוסיף קומפוננטה

אם רוצים להעביר מידע מתוך TS צריך להוסיף @INPUT לפני

צריך לעבור על החוברות

צילום בטלפון על הפרוייקט לקראת שיעור 2



**שיעור 2**

**31.12.19**

**הגדרת מודול**

Appmodule בעיקרון לא צריך להכיל שום קוד .

Bootstrap – מה הקומפוננטה הראשית שתרוץ בהתחלת הישום .

Decleataions – מערך של קומפוננטוות ועוד..לא services

Imports - מערך של מודולים

Exports - רק מה שמופיע פה יהיה מוכר מחוץ למודול (מתוך ה-declarations), וגם למודולים(שיחשוף גם את הקומפוננטות של המודול)

קומפוננטות בהגדרה רלוונטיות רק למודול עצמו (internal)

**עדיף להגדיר מודול ותיקיות עם קומפוננטות כי ככה אפשר לטעון כל מודול בנפרד וככה יהיה זמן טעינה יותר מהיר .**

אם רוצים ליצור interface , רצוי לשים אותו בתייקיה יעודית לinterface.(models)

משתמשים היום במקום ב-table ב- DIV עם CSS של grid (CSS3)

Display:grid

Grid-template-rows: 80% 20%

Height:100vh – קובע שגודל האתר שלנו כגודל הדף שנפתח

Scss מקומפל לcss

אפשר ב-scss לרשום הגדרה בתוך הגדרה .

בTS :

בconstructor – רצוי מאוד שיהיה ריק

רצוי להריץ פונקצית ואיתחולים בתוך ngOnInit , ככה הhtml ירוץ מיד אחרי זה .

Display:flex – חלוקה בצורה שווה במרווחים

אפשר להשוות סטטוס תשובה עם מחרוזת כדי שיהיה קריא יותר

נגדיר type שיכול להכיל מספר אפשרויות

Export type aaaa =’g’|’f’|’r’

אפשר לבחור רק ב3 אפשרויות לAAAA

או פשוט לכתוב שם משתמש שאפשר ממנו לבחור רק מחרוזות מסויימות

Familystatus:’single’|’married’|……..

[ngClass]

CSS מותנה בתוך קומפוננטה

בתרגיל שלנו של השאלות , גם השאלון צריך לדעת האם השאלה נכונה , גם הקומפוננטה של השאלה עצמה וגם הקונטיינר כדי לדעת מה לעשות (לצבוע)

ככה יצרנו את אותה פונקציה ב-3 מקומות שונים .

איך פותרים את זה?

צריך משהו אחד שיקראו לו 3 פעמים

ניצור תיקייה של services

Ng g s name

Name.service.ts

מגדירים class

נוסיף את הפונציה פה

בכל מקום שאנחנו רוצים לקרוא לפונקציה נקרא לCLASS מתוך הבנאי (constrature).

אנחנו נקרא

Constractor (**private** nameofservice : service)

צריך להגדיר הservice בproviders רצוי ב- appmodule

Provides :[{provide: whenAskedname ,useClass :use-this-class}]

ככה אפשר בכל מקום במערכת שלנו שנרצה לקרוא לפונקציה שעושה משהו אנחנו נקרא לwhenaskedname אבל מה שירוץ זה מה שכתוב אחרי ה-useClass (תרוץ הפונקציה בתוך use-this-class)

אנחנו גם נשים את כל הנתונים בתוך ה-service ואנחנו נקרא מתוך ה-list ואז אנחנו נקרא ל-service כדי לקבל את הנתונים ובכל "נוריד" אחריות מ-LIST לגבי הנתונים ובעתיד נוכל לשנות בקלות מאיפה נקרא את הנתונים.

**אפשר אבל לא רצוי**

אפשר להרשם ב-appmodule מתוך הclass שלנו ע"י injectable מעל איפה שאנחנו משתמשים...

**PIPE**

הצגת משתנים של ה-TS בתוך ה-html אבל בצורות שונות .

זה כמו overrite אבל רק לתצוגה , מניפוצליה של תצוגה בלבד.

מסומן ב-|

יש פיפ קיים ויש כזה שניצור

קיים למשל

{{title |uppercase}}

אפשר גם

{{birthdate|date:”dd-MM-yyyy”}}

תצוגת יום הולדת בצורה שאנחנו רוצים

יצירת PIPE

Ng g p suffix

אנחנו רוצים כל מספר גדול מ-999 שיעשה את זה כ-1K

הגדרת הפיפ בתוך declarations

בתוך ה-pipe יש פונקציה של transform שהיא הפונציה שרצה

הפוקנציה מקבלת ערכים ומחזירה מה יריץ

{{stringTochange|pipe-name:parm1:parm2…..}}

תרגילים:

1. פיפ שיכול לקחת מספר ולהמיר אותו לאלפים במידה ויש.
2. מקבל דקות רושם כמה שעות זה וכמה דקות (בלי אפסים )

אנחנו לפעמים רוצים שכמה קומפוננטות ישתנו במקביל ורוצים שכל הקומפוננטות ישתנו במקביל.

Reactive programing

שרשימת השאלות משתנה אנחנו רוצים טריגר שיודיע מתי משתנה.

` - הגרש ליד ה-1 נותן אפשרות לרישום מחרוזת עם משתנים

`${this.name} + ‘’hello’`

איך הטריגר קורה?

אנגולר מעדכן מסך ב-3 מקרים:

1.שינוי שקרה בUI , קליק , כתיבה לטקסט

2.setinterval settimeout מופעלות , משנה את המסך

3.תשובה של בקשה של http

אנחנו צריכים לשנות נתון בתוך service ולהודיע לcomponenet .

נשתמש בevent מיוחד .

נגדיר בתוך ה-servive

New subject<type of param>

ונריץ את הפונקציה היעודית שלו עם הנתון אותו אנחנו צריכים לשנות

בקריאה של הEVENT אנחנו נשתמש ב-NEXT (במקום EMIT של output)

ואז בקומפוננטה רושמים subscribe

ככה נרשמים לתוך האירוע שרץ בתוך ה-service

נרשמנו מעכשיו לservice אבל מה קורה אם אנחנו כבר בפנים ורוצים ישר תגובה.

**רוצים את התשובה האחרונה שהודיעו .**

במקום subject נרשום behaviorsubject (שהוא הנפוץ ביותר!)

**אנחנו תמיד נרצה להשתמש בbehaviersubject**

**Subject\behaviorsubject מקביל ל- eventemitter**

Replaysubect מריץ את כל התשובות מאז ההתחלה. לא נפוץ

רישום הsubscribe יעשה כך:

onInit

Sub subsctiption = this.service .function.subscribe(מה עושים

ואחרי זה אנחנו צריכים להעיף

**OnDistroy**

**Sub.unsbscribe(subscriotion)**

אחרת יש לנו זליגת זיכרון כי האובייקט ישאר מחובר אל מול הservice שלנו .

צריכים להוריד רישום מאובייקטים שאנחנו יצרנו .

Rxjs - היא ספרייה שממנה אנחנו מוציאים את ה-behviorsubject ויש עוד דברים שם ...

1.observable

2.subject ,behaviorsubejct

ל-1 יש רק subsribe

ל-2 יש

Subsribe , next , complete,error

בגלל שיש לbehaviorsubect הרבה כוח , שכל אחד ממי שרשום לservice יכול לעשות next או complete או error , אנחנו לא נרצה לחשוף אותו החוצה.

במקום זה אנחנו נעשה אותו משתנה private ונשנמך אותו לאובייקט ovservable כדי שיוכל לעשות רק רישום ואנחנו בתוך הservice נעשה את הnext .

במקום להרשם וגם להוריד את הרישום (בגלל דליפת זיכרון) אפשר אפשרות אחרת

Name$ :observebale<number>;

ובקוד של ה-html

{{name$| async}}

זה חוסך לנו גם את הרישום וגם את ההסרה.

זה במקרה שאנחנו רוצים רק שינוי של UI אז אנגולר חוסכת לנו בקוד הנ"ל.

שיעור שלישי(2.1.2020)

Observable - ניתן להאזין בלבד.(ע"י subscribe)

Subject\behviorsubject –ניתן להפיץ וגם להאזין יש כמה פונקציות next error complete .

כל הפעולות האלה הם א-סינכרונית , יש פונקציות שאנחנו צריכים לשלוח וזה עובד מתי שהתהליך בשרת הסתיים.

אנחנו נשתמש באובייקט promise

אם רוצים פונקציות א-סינכרוניות אז נקרא לפונקציות then ( הצלחה ) catch (כישלון )

רצוי להחזיר תמיד promise ולהפוך כל service לא-סינכרוני ולהתייחס לזה בהתאם.

לא כדאי להשתמש באופציה הזאת למעלה , כי אנחנו לרוב נרצה לחכות לפעולה מסויימת בתוך הפונקציה בלי לתקוע את הGUI.

**הכי כדאי לכתוב על פונקציה async ואז פעולה בתוכה אפשר לעשות לידה await שזה אומר שמחכים עד שהשורה תסתיים וזה לא "תוקע" את הפונקציה שקראה לנו .**

**הפונקציה כמובן צריכה להחזיר promise**

אפשר להשתמש בpromise.all ולחכות לכל הpromise עם await .

**מה נמצא בספריית rxjs**

Observable<type of variable>

כאשר משתמשים ב-service כמו למשל observable אנחנו יכולים להשתמש ב-pipe שיכול לעשות מנפולציה על הנתונים הנכנסים .

נניח :

Pipe(filter ( פה עושים פעולה על הנתונים שנחליט האם להחזיר או לא להחזיר)

אפשר גם להשתמש בmap

Pipe (map( מניפולציה על נתונים אבל מחוייב להחזיר ערך שהגדרנו באובזרבבול

אפשר

Pipe(deboundtime

זה במקרה ואנחנו רוצים נניח להביא רשימת אנשים אבל רוצים לחכות קצת כדי לראות האם המשתמש עדיין מקליד תווים.

httpclient הוא אובייקט מובנה שאפשר לעשות בו פרוטוקולים מוכרים

מגיע מתוך @angular/common/http

יש אתר שנקרא json to typescript

אפשר לשפוך לתוכו את הjson ולקבל מבנה נתונים של TS.

צריך להגדיר בapp.module בתוך imports צריך להוסיף httpclientmodule

פונקצית get מחזיר observable של המערך אותו אנחנו רוצים.

Httpclient יודע האם מישהו נרשם ורק אז הולך לשרת .

צריך להבדיל בין filter שמגיע מתוך ה-http שמביא נתונים על פי מה שהגדרנו כפילטר , ויש פילטר שעובד על הנתונים עצמם כאשר הם כבר הגיעו.

אנחנו מריצים npm I –g json-server

הוא מתקין שרת rest מקומי

הוא יכול לעבוד אל מול קובץ JSON

Json-server שם קובץ

אנחנו צריכים להריץ רשימת מוצרים ע"פ הקובץ הנ"ל

(הקוד המלא הדומה נמצא בתוך day3 -shows-app)

Extention עבור vscode

Angular files - מאפשר בקליק ימני על תיקייה להוסיף כל סוג אובייקט (קומפוננטה , סרוויס וכו')

**טפסים**

אנחנו רוצים למנוע מצב של databinding דו כיווני , כיוון שהתפקיד של הטופס בתוך ה-html יהיה שונה ממסד הנתונים.

שעובדים עם טפסים צריך להוסיף בapp.mdules וגם בטפסים .

reactiveFormsModule

FormModule

בתוך ה- imports

יש כמה אובייקטים :

Formgroup ,Formcontrol

Formarray

צריך לקחת שדה מתוך ה-TS לחבר אותו לcontrol ואז לחבר בניהם

Fromgroup רשימה של fromcontrol

כל formcotrol = משתנה של ה-TS

צריך להגדיר formgroup (@anguler/formgroup)

Formbuilder – מאפשר לבנות טופס (מאתחלים אותו בconstrctor בתוך private)

Showform : FormGroup ;

Constructor(private fb :formBuilder )

This.showform.fb.group(שמות שדות עם []

אפשר לעשות

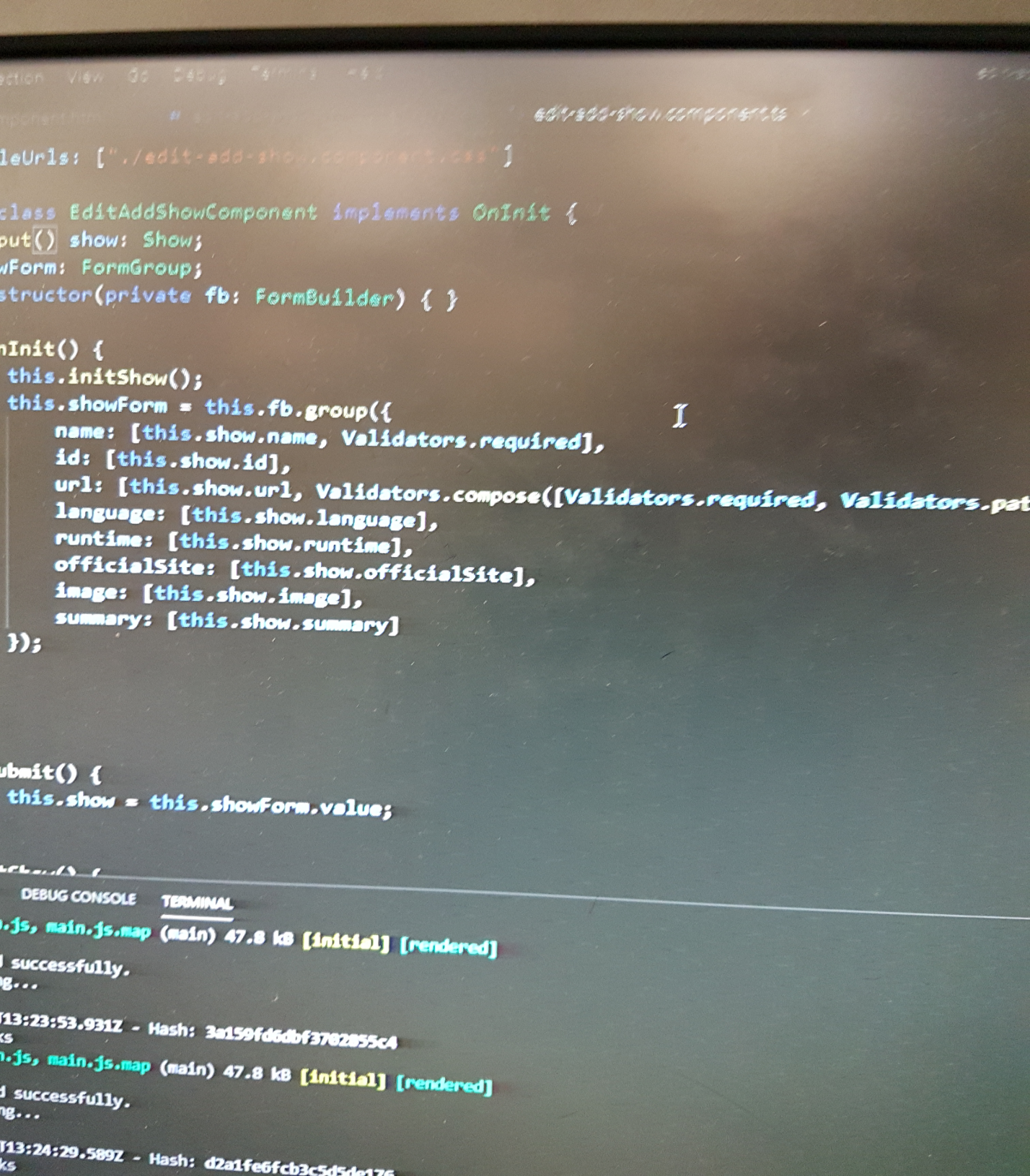
Fieldname |json

מאפשר להריץ json בצורה יותר נעימה

השאיפה שלנו ליצור מבנה בתוך ה-TS ואז form עם מבנה מקביל בשביל הלוגיקה

Showform.invalid

נותן לנו האם יש וולידציה של המשתנה



Onchanges בהגדרה של הקומפוננטה implements עובד כאשר יש שינוי בinputs , יש לו גם פרמטר אופציונלי שנקרא simplechanges שאומר בדיוק איפה השינוי נעשה ואיפה בדיוק באובייקטים שיש לנו בפנים

כל פעולות קבלת הנתונים יקבלו observable כל שינויים יקבלו promise כדי לקבל את await

**שיעור 4 (6.1.2020)**

**חזרה על החומר**

EventEmiter – הודעה על אירוע שקרה , קומפונטטה אחת רוצה להודיע לקומפוננטה אחרת משהו קרה.

פונקציה בSERVICE תמיד תחזיר Promise או observable. (הכי כדאי לא מחייב כמובן)

Promise מסתיים בפעם אחת , לא יודעים מתי אבל יסתיים פעם אחת.

GET – select – behviorSbject (אפשר לעשות מניפולציות על התוצאה כמו פילטר.

SET – delete update insert jobinSQL – נשתמש ב-PROMISE (הפעולה הצליחה הפעולה נכשלה)

במקרה של promise אנחנו יכולים להשתמש בפונקציה ב-async וawait

הAWAIT יודע לחכות לprimose.

**חומר חדש**

כאשר אנחנו רוצים לכתוב אפליקציה בתוך הארגון , בתוך ה-html נריץ בזמן ריצה <base target=”/app1”

אנחנו רוצים לעשות כמה דפים ולא הכל על אותו הדף.

איך עושים? SPA –single page application.

Back בתוך דפדפן עובד כל פעם שאנחנו משנים URL .

היום יש historymode במקום #ANCOR שהיה קיים פעם .

בתוך הדף יש

<base href=’gggggg’

ככה הדפדפן לא ירפרש.

עוד משהו שאנחנו צריכים להגדיר , זה את האיזור שאנחנו רוצים להחליף.

Router-outlet- אחת לכל מודול

תגית שהיא מתחלפת (מקביל לASPX שנמצא בתוך הMASTER PAGE - contentpageholder

אנחנו נגדיר "טבלה" שתגיד מתי שאנחנו רוצים X איזה קומפוטטנה נקבל.

איך נגדיר router-outlet ?

routerModule נמצא בתוך @angular/Router מוגדר בappmodule – בimport

נגדיר גם קובץ חדש module.routing עם הטבלה שדיברנו עליה למעלה (אם לוחצים על X מה נריץ)

ושם נרשום את כל החוקים (אם נלחץ על X איזה קומפוננטה אנחנו נעבור), ובappmodule נרשום בתוך ה-imprts inroot(params

Routerlink- דואג שלא יתרפרש הדף ועדיין יתעדכן הנתונים .(בתוך תגית A)

<a routerlink=’/home’>

כאשר אנחנו רוצים לנווט מתוך כפתור , נגדיר בתוך ה-קונסטרקטור router

ובפנקציה שנריץ

This.router.navigate[‘/home’]

Routerlink מאנגולר 7 אפשר לעשות על כל אלמנט של HTML.

Routerlinkactive=CSSCLASS –CSS שירוץ כאשר אנחנו תואמים בURL הנוכחי לאיפה שרצינו להגיע.

איך מעבירים פרמטרים בURL

Zap.co.il/products/hp/aaa?x=3#g

Urlparm

Quarysting param

Hash

Hash – פרמטר בודד אחד שאפשר להשאיר אופציונלי.

Quarystring – אופציונלי – הדף אמור לתפקד גם בלי הנתון הזה.

בטבלת ההמרה של router אם נרצה לשלוח פרמטר אחרי ה-path נשלח :xparam על כל פרמטר שנשלח אחרי ה ?

טבלת הפרמטרים תהיה ככה :

Export const APP\_ROUTE:route []=[

{path:’home , component : componenetNAme },

{path:’products/:parmx/:parmy ……..

];

איך קוראים לפרמטרים?

נקרא בקונסטרקטור ל-activateroute

הפרמטרים הם תמיד string

ואז לקרוא לפרטרים

activeRoute.snapshot.params.paramx

**תמיד כדאי להשתמש ב**

activeRote.params.subsribe(params=>{this.sum = +params.Xparam \* +param.yParam

**ה-+ הופך את הstring למספר.**

**כל מה שכתבנו מתייחס לURLPARM!**

עכשיו נשתמש ב-quarystring

This.routerobj.naviagte([‘url’,urlparamx,urlparamy ],{quaryparam:{oprator:paramname})

ואז נרשם ל-

This.activeRoute.quartparam.subsribe(….

אנחנו נרשמים גם לURLPARMA וגם ל QUARYSTRING.

2 רישומים , אולי אפשר אחרת?

רוצים להגיב לכמה אירועים (observable)

יש combinelatest והוא יגיד לנו את הערך האחרון של כל רישום.

Combinetatest ([this.avtiveteroute.urlparam , this.avtivatroute.quaryparam]).subsribe([urlparam,quaryparam)= >

{

Const x= urlparam.xparam

Const op = quaryparam.op

}

Const xxx = paramX || paramy –אם אחד קיים תשמש בו ואם לא תשמש בשני

אפשר לרשום בצורה אחרת :

This.Sum$ = Combinetatest ([this.avtiveteroute.urlparam , this.avtivatroute.quaryparam]).pipe(map(

urlparam,quaryparam)= >

{

Const x= urlparam.xparam

Return +eval(xparam\*yparam

מקבלים OBSERVABLE של מספר .