json <http://voucherservice.integrations.at/>

LPE – Long Path Eraser Tool

typescriptlang.org

Strg +ö 🡪 öffnet console in visual code

definitelytyped.org ->

strg + p 🡪 >Format Document 🡪 Formatiert document

Chrome 🡪 F12 = Console

# 01 TypeScript:

# install yarn

Npm i –g yarn

#installiert dependencies

Yarn

# projekt starten und im browser anzeigen

Ng serve –o true

bundle: optimiertes java script code

[root]/src/app

* .html, .ts, .spec.ts (unit testing), .css

Chrome console:

webpack: beinhaltet folderhierarchie wie im VS

webpack-internal: beinhaltet schon fertig JS code (Abhängigkeiten aufgelöst)

strg + p -> filename – um file zu öffnen

# 02 Angular:

# Projekt builden

# Files werden in dist abgelegt und beinhalten \*.js Files und \*.html

ng build

Bei Visual Studio Code muss node.js nachinstalliert werden [www.nodejs.org](http://www.nodejs.org)

#Dann muss Angular CLI installiert werden

npm install –g @angular/cli

#neues Projekt erstellen

ng new

# neue Component (dry run) = ng generate component

ng g c Nested --dry

#zeigt installierte extensions an

code –list-extensions

# 03 Fundamentals:

{{expression}} -> wird im view (.html file) ausgewertet

Angular Augary 🡪 Chrome extension Angular debugger tool!

<div \*ngIf="person?.name">

{{person | json}}

</div>

person?.name = nullable operator auf name, existiert name?

person | json = person als json ausgeben

#erstellt in subfolder

ng g c accounts/acountlist

# für releases genau dependencies mitgeben

shrinkwrap

# 04 Routing:

# 05 UI:

scss – sass css 🡪 neue art von css, programmierbar (variablen)

#CSS wird nicht an child vererbt

umgehen indem man:

* im child encapsulation auf none setzt
* als global style definieren

Material:

* richtigen module importieren (app.modules.ts)

# 06 Data Acess: