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:

# wie yarn oder npm install

dotnet restore

dotnet run

dotnet watch run # reagiert auch live auf changes

#logger für javascript errors

jsnlog

#DB first mit EF

dotnet ef dbcontext scaffold [connectionstring]

#änderungen im code um zb zeilen hinzuzufügen 🡪 dann

dotnet ef migrations add AccountUpdate

dotnet ef database update

# API development test tool

postman

# project generator (da von usern, oft veraltet)

yeoman

#angular bietet zwei http clients: http (alt) & httpclient

http client liefert per default die daten, mit observe: "response" werden dann auch status code usw retourniert

#firebasedb is nosql db von google

# 07: Reactive Programming

promise vs observable

#framework, rxjs wird mit angular automatisch mitinstallier

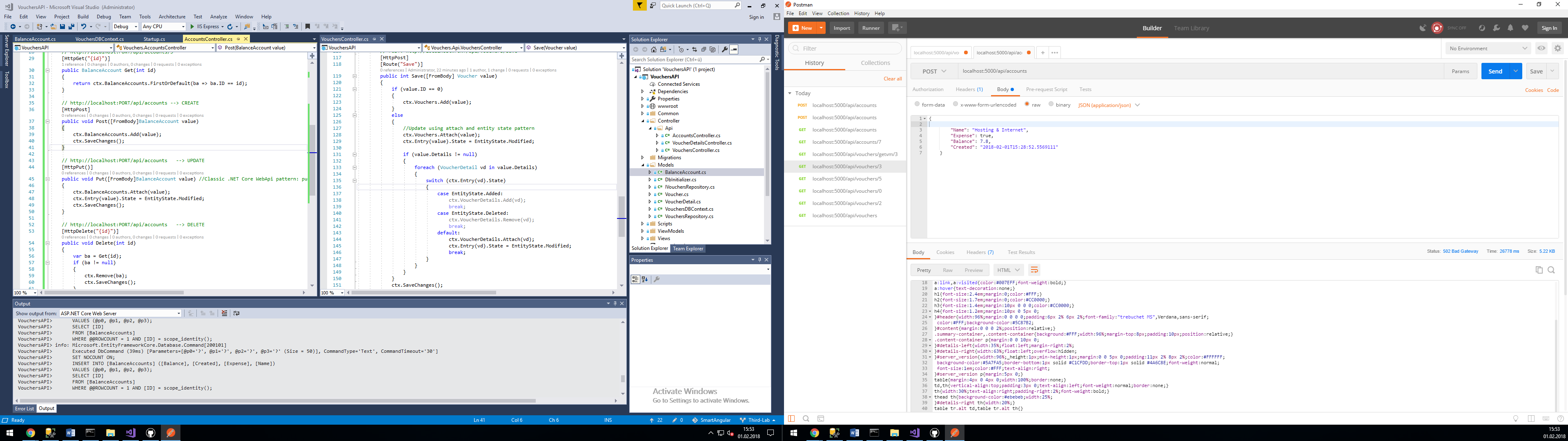
reactivex.io

#Async pipe – wird bei observables verwendet

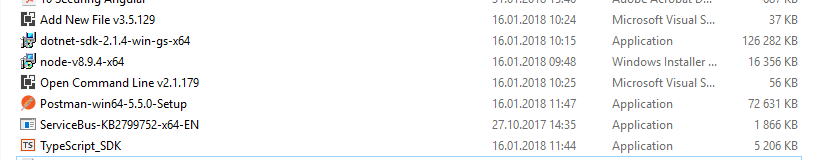
nbrOBs | async

# 08: Forms

#empfohlen reactive formbuilder, ist kürzer und unit testbar



DOWNLOADS:



#datastore

ngrx

#linq – language integrated query, wird im API verwendet um auf daten aus DB zuzugreifen

#linq gibt es auch für typescript

#authentifizierung

jwt token zb mit firebase

#hole alle nuget packages

dotnet restore

#Alexander Pajer (Trainer) github

<https://github.com/ARambazamba>

karma = testrunner

jasmine = testing framework

#build angular

ng build –prod

#build core

dockerize:

* syntax hiughlight und docker explorer extension in vscode