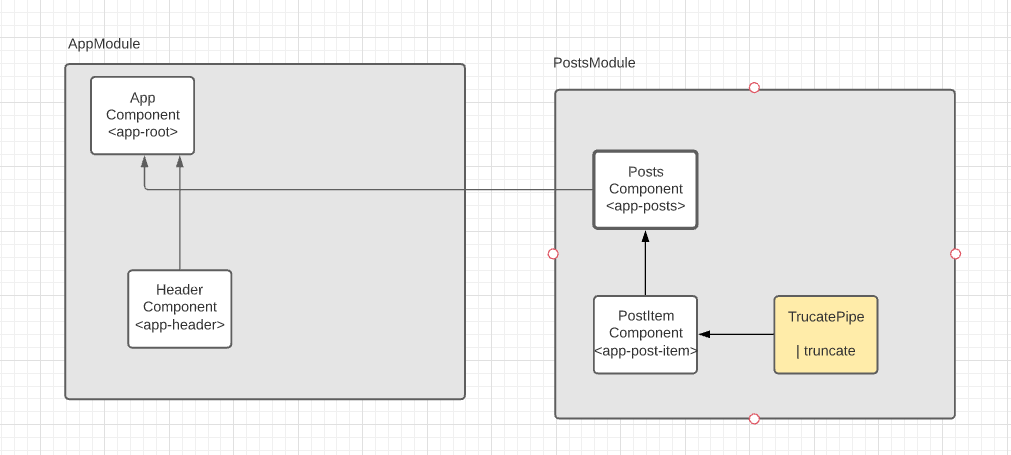
Angular simple blog

(v3 update)

## Feature modul szervezés

* A “post”-ok, mint “feature” megjelenítésével kapcsolatos komponenseket szervezzük ki egy új modulba!
* A module neve: “posts.module.ts”, generáld CLI segítségével!
* Új modulhoz új mappa jár és a modulhoz tartozó elemeket a modul mappájába szokás tartani.  
    
  A modulhoz tartozó komponenseket (module, component, pipe és egyéb elemeket a posts ) mozgasd át a “posts” modul mappájába!  
    
  Használhathoz:  
  - exportáld a PostsComponent-et a posts.module-ban  
  - Importáld a PostsModule-t az AppModul-ban → így már használható a <app-posts> komponens szelektorját az AppComponent template-jében  
    
    
  

## **2. feladat: PostsService** Jelenleg a post elemek tömbje az AppComponent-ben van, ami nem tl előnyös. Az adatokkal kapcsolatos logikát service-kben érdemes táolni, ezért a jelenlegi “mock” post elem tömböt tegyük át egy PostsModule-ban létrehozott service-be Élesben: 1. Készíts egy új services-t “PostsService” néven! 2. PostService implementálása: - van egy getPosts metódusa, ami visszatéríti a posts: Posts[] tömböt (ami jelenlega AppComponent-ben van). - modulban regisztráld a service-t (cli automatikusan megtette, ha cli-vel generaáltad!) 3. PostsComponent felkésítése/átalakítása

Jeleneleg PostsComponent egy inputon keresztül várja a a Post[] típusú adatot, de ehelyett szeretnénk az új service-t használni!   
  
- injektáld a service-t a posts.component construktoában!  
  
- a komponens ngOnInit lifecícle metódusában hívd meg a service   
  
getPosts metódusát, ami visszadja a posztok tömbjét   
- mentsd el a service által visszadott tömböt a komponens mezejében, ami eddig inputként szolgált.  
  
 @Iinput dekorátorra már nem lesz szükség!   
   
- A posztok kilistázása ugyanúgy működjön, ahogy eddig.  
  
Konklúzió: most már az adatok a service segítségével kapja meg a PostsComponent, ezáltal egy smart vagy container komponens lett!

# 