

MATEMATIKAI ÉS INFORMATIKAI INTÉZET

Angular mobil framework bemutatása

Készítette

Gór Milán

PTI BSC Lev.

EGER, 2019

Tartalomjegyzék

1. Előkészületek	4
1.1. Szükséges szoftverek	4
1.2. A szoftvereket telepítése	4
1.3. Kód szerkesztése	4
1.4. Projekt készítése	5
1.4.1. Létrehozás	5
1.4.2. Elindítás	5
2. Az alkalmazás	6
2.1. Home automation	6
2.1.1. A terv	6
2.1.2. A megvalósítás	6
2.1.3. Kommunikáció	6
2.1.4. A keretrendszer	7
2.1.5. A callback	7
2.1.6. Az alkalmazás	8
2.1.7. Github	8

Bevezetés

A dokumentum célja, bemutatni az **AngularJS** és a **NativeJS** által közösen létrehozott keretrendszert melyet a köztudatban csak **Angular Mobile Framework**ként említenek. A keretrendszert azért hozták létre, hogy platform függetlenül lehessen mobil applikációkat létrehozni, és egyszerűen lehessen az applikációknak grafikus felületet készíteni. Az egész keretrendszer **Javascript** alapokon fekszik. Régóta probléma a mobil applikáció fejlesztésben, hogy kevés az olyan nyelv amivel a nagy mobil operációs rendszerekre (iOS, Android) egyszerre tudnánk lefordítani a megírt alkalmazásunkat. Az **Angular Mobile Framework** ebben nagy segítség, hogy a megírt alkalmazásunkat könnyedén tudjuk fordítani majd futtatni **iOs** valamint **Android** készülékre probléma és bármiféle átalakítás nélkül. Ez nagyon nagy segítség lehet, főleg az olyan alkalmazásoknál melyek "csak" adatot dolgoznak fel és továbbítanak, vagyis kevés **GPU**-t kell használniuk, és lényegtelen ilyen szempontból a sebesség, mert nem fogjuk észrevenni a 4 és 8 magos készülékünkön azt a plusz réteget amit ez képez.

A dokumentum célja, hogy röviden és tömören de célravezetően bemutassa a keretrendszer használatát valamint előnyeit és hátrányait. A keretrendszer használatához szükségünk lesz pár szoftver telepítésére, de erről majd az előkészületek fejezetben részletesebben beszélünk.

1. fejezet

Előkészületek

1.1. Szükséges szoftverek

Ahhoz, hogy az Angular Mobile Framework tökéletesen fusson a gépünkön, előre telepítenünk kell pár szoftvert. Ezek a szoftverek úgynevezett **Cross Platform** szoftverek, ami annyit jelent, hogy operációs rendszertől függetlenül telepíthetjük őket, vagyis az összes nagy és népszerű (MacOs, Linux, Windows) operációs rendszerre gond nélkül fog települni.

1.2. A szoftvereket telepítése

A következő szoftverek telepítése szükséges:

- nodeJs
- nativescript CLI
- JDK 8
- Android SDK
- playground app (telefonra)

1.3. Kód szerkesztése

Bármilyen szövegszerkesztőt használhatunk a kódunk írásához, de én leginkább a **Sublime Text 3**-at ajánlom mindenkinek, mert nagyon jól paraméterezhető és sok hasznos kiegészítő található hozzá.

1.4. Projekt készítése

1.4.1. Létrehozás

A következő parancsot kiadva tudjuk létrehozni első projektünket:

```
tns create projektneve --template tns-template-blank-ng
```

1.4.2. Elindítás

Indítsuk el az Nativescript viewer alkalmazást a telefonunkon, és adjuk ki a következő parancsot:

```
tns preview
```

A fent megadott parancsot kiadva a terminálban megjelenik egy QR kód melyet a telefontal leolvasva tudjuk az alkalmazásunkat megtekinteni a telefonon.

2. fejezet

Az alkalmazás

2.1. Home automation

2.1.1. A terv

Régóta tervben volt egy automatizálási projekt, amivel különböző dolgokat akár távolról is lehet vezérelni a lakásban. Az ötlet onnan jött, hogy sokat utazunk vidékre és télen vagy nyáron jó lenne nem a hideg vagy a meleg lakásba hazaérkezni, fáradtan.

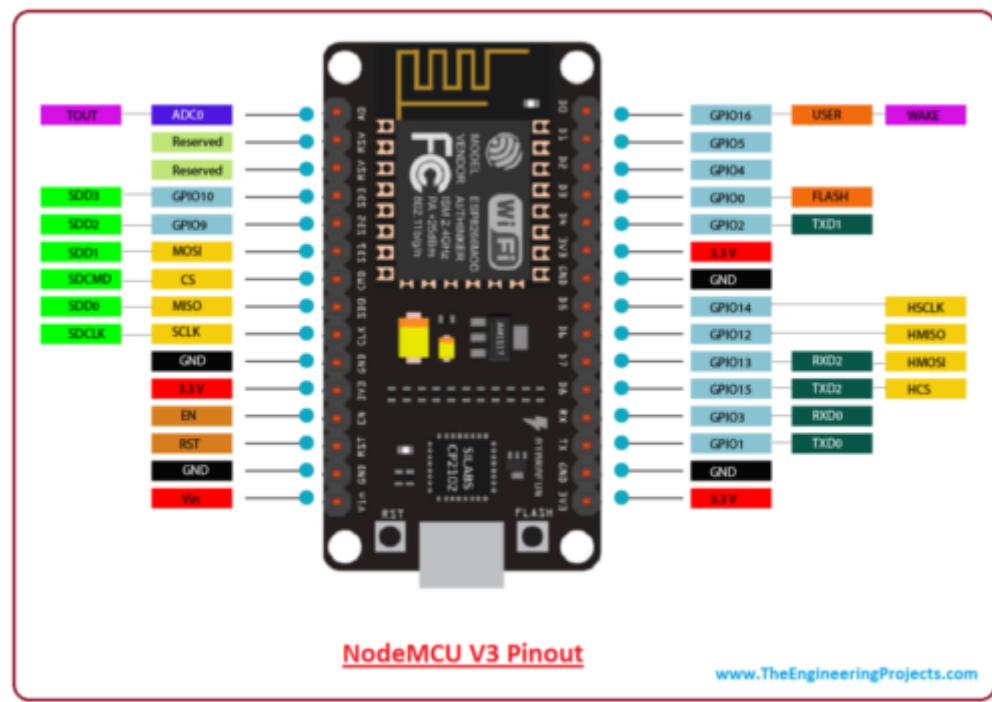
2.1.2. A megvalósítás

Az alkalmazás http kérésekkel vezérel egy **NodeMCU** nevű **ESP8266** processzorral ellátott fejlesztői panelt.

2.1.3. Kommunikáció

Mivel ez a panel wifi kapcsolattal rendelkezik ezért a legegyszerűbb és legjobb dolog írni hozzá egy applikációt, és wifin már kommunikálhatnak is a belső hálózaton. A lekérdezések mind **GET** metódussal a beállítások pedig **POST** metódussal kerülnek küldésre. A kommunikációnak nincs szabványos protokollja, viszont az adatok küldése és fogadása **JSON** adatformátumban történik. A **JSON** azért népszerű mert nagyon egyszerűen lehet belőle objektumot készíteni, a struktúrája is nagyon hasonló. JavaScriptben például majdnem ugyanaz, ezért egyszerűen készíthető az ilyen adatformátumból kész objektum amit már használhatunk is a kódunkban. Ha hiba nélküli válasz jön annak a státusz kódja 20x, ha pedig bármilyen hiba lépne fel a kérés során akkor az eszköz 40x típusú hibakóddal tér vissza.

2.1. ábra. Egy NodeMCU láb kiosztása



2.1.4. A keretrendszer

A **NativeScript** és az **Angular** segítségével nagyon gyorsan lehet jó minőségű alkalmazást készíteni. Elemek listázása az Angular segítségével:

```
<GridLayout class="page">
  <ListView [items]="items" class="list-group">
    <ng-template let-item="item">
      <Label [nsRouterLink]="['/item', item.id]"
        [text]="item.name" class="list-group-item">
    </Label>
  </ng-template>
</ListView>
</GridLayout>
```

A szoftver JavaScript nyelven íródik és az Angular sajátos HTML nyelvét felhasználva (XML-re épül) nagyon jó dolgokat lehet elérni.

2.1.5. A callback

Amit nehéz lehet megszokni az az, hogy a JavaScript miatt szinte minden hívás aszinkron, de ha megszokjuk, hogy adat nem akkor jön amikor szeretnénk hanem majd egyszer csak ott lesz, de aggodalomra semmi ok, mert minden event-ről értesülni fogunk amire

2.2. ábra. Az alkalmazásról egy képernyőfotó



feliratkozunk.

2.1.6. Az alkalmazás

Nagyon sok része már teljesen készen van, de vannak még hiányosságok. A következő dolgok fognak a jövőben még belekerülni:

- hőmérséklet vezérlés
- eszköz UDP-n megkeresése
- a szenzorok adatainak mentése szerverre

2.1.7. Github

Az általam készített kód megtalálható a github-on on a következő címen lehet elérni:
<https://github.com/gorcsek/homeautomation-app>

Irodalomjegyzék

nativescript.org *Official site*. Native Script dokumentáció

nodejs.org *Official site*. NodeJS dokumentáció

stackoverflow.com *Community*. A fejlesztők fellegvára

theengineeringprojects.com *Community*. Hasonló projektek tárháza