

# Szótár

Kliensoldali technológiák házi feladat – Készítette: Nagy Péter Ákos (G8N2W3)

## Összefoglaló

A szótár alkalmazás egy egyszerű fordító és szinonima alkalmazás, amely a Yandex API segítségével közel 30 nyelet ismer. A felhasználó kiválaszthatja a forrás- és célnyelvet, és a beírt szónak az alkalmazás kiírja a lehetséges fordításait (ha a forrás- és célnyelv azonos akkor a szinonimáit). A fordításokról (vagy szinonimákról) további adatot is megtudhatunk, ha rákattintunk a fordított szóra. A szótár használatához a felhasználó meg kell nyissa az oldalt, amely a fejlesztőknek a <http://localhost:4200> címen érhető el.

**Szótár** Fordító

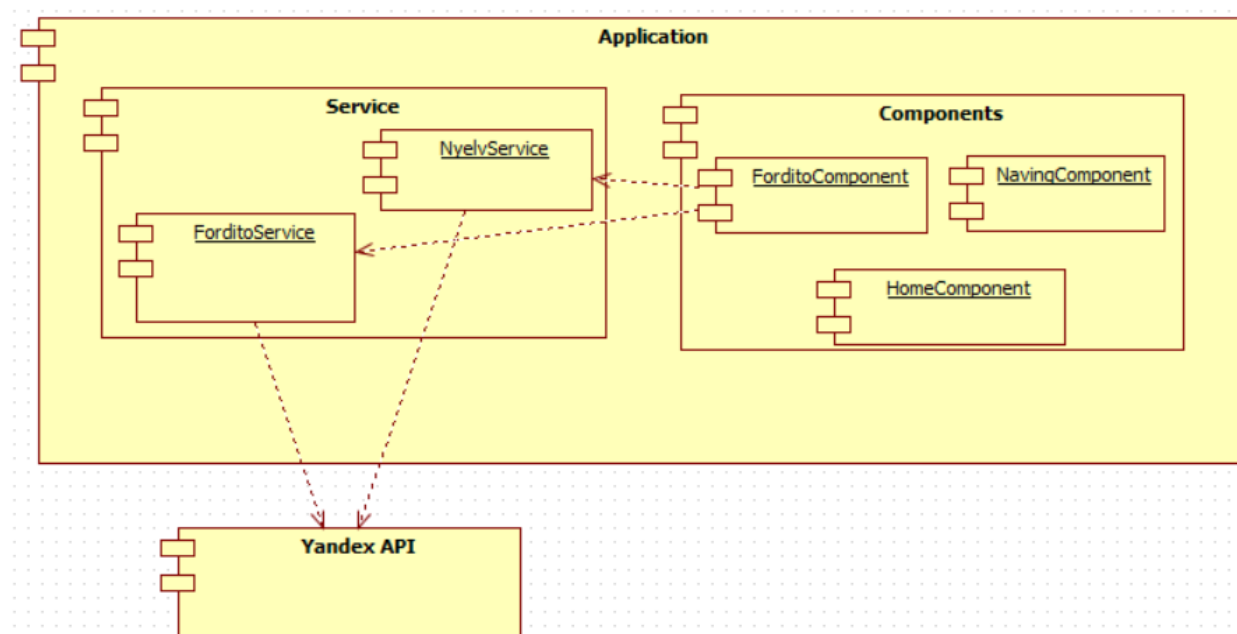
### Fordítás

Forrásnyelv:	Magyar ▾
Célnyelv:	Magyar ▾
Fordítandó szó:	<input type="text" value="ló"/> <b>Fordít</b>
Fordítás:	<ul style="list-style-type: none"><li>paripa</li></ul>

### Részletek

Szó:	paripa
Nyelv:	Magyar
Szófaj:	főnév

## Architektúra



## Komponensek

- ForditoComponent : a program azon része, mely a fordítás és szinonima találás/validálással foglalkozik
- NavingComponent : a navigációért felelős komponens
- HomeComponent : a kezdőlap komponense, egy Welcome képet jelenít meg

## Szolgáltatások

- ForditoService : a fordításért felelős szolgáltatás
- NyelvService : a létező nyelvpárosítások ( azaz, mely nyelvekről melyekre lehet fordítani) lekérdezésére

## Fontosabb egységek

### ForditoComponent

Ez a komponens felelős azért, hogy megjelenítse a megfelelő adatokat a fordítás során: mely forrásnyelvek ismertek, a kiválasztott forrásnyelvhez milyen célnyelvek ismertek. A komponens validálja is az inputot, tehát amíg nincs cél- és forrásnyelv kiválasztva, addig a felhasználó nem kattinthat a „fordít” gombra, mely megjeleníti a fordítás(oka)t. Ha az API nem talál semmilyen fordítást a beírt szóhoz akkor a fordítás eredményei helyén a következő üzenet jelenik meg: „Nem találtuk a fordítást”. Miután megjelentek az esetleges fordítások, a szavakra rákattintva, részleteket jeleníthetünk meg a szóról: megjeleníti az eredmény nyelvét, így a felhasználó is tudja ellenőrizni, hogy valójában a kívánt nyelvre történt a fordítás, valamint a szófaját is megjeleníti, így a többértelmű szavak esetén egyértelművé válhat, hogy melyik fordítás melyik jelentéshez tartozik, ahogy az alábbi képernyőképeken is látható.

#### Fordítás

Forrásnyelv:	Magyar
Célnyelv:	Orosz
Fordítandó szó:	vár <span>Fordít</span>
Fordítás:	<ul style="list-style-type: none"><li>• замок</li><li>• подождать</li></ul>

#### Részletek

Szó:	замок
Nyelv:	Orosz
Szófaj:	főnév

#### Részletek

Szó:	подождать
Nyelv:	Orosz
Szófaj:	ige

### NyelvService

A Yandex API-tól lekért adatok alapján létrehozza az ismert forrás- és célnyelv párosításokat. Az API stringekben válaszol a kérésekre az alábbi formátumban : { „hu-ru”, „ge-en”}, melyeknek jelentése, hogy az API képes magyarról oroszra és németről angolra fordítani. Ezen string párosok alapján egy Nyelv modellű objektumban létrehozza az összes ismert forrásnyelvet, és minden nyelvhez tárolja, hogy ez

milyen további nyelveket ismer, azaz, hogy mire lehet az adott célnyelvről fordítani. A „hu-hu” alakú stringek jelentése, hogy tud magyarul szinonimákat találni az egyes szavakhoz. Mivel az API kérés időigényes művelet, ezért az első lekérés után az ismert nyelvpárokat a böngésző localStorage-ában tárolja, ezzel is növelve a teljesítményt. A rövidítésből a Nyelv objektum előállítását a *createNyelvek()* függvény végzi, a *roviditesNyelvOsszerendeles* map típusú adatstruktúra segítségével, melyben minden rövidítéshez egy magyar nyelvű teljes nyelv név tartozik pl. („ru”, „Orosz”). Ezek az összerendelések úgy mond „hardcodolva” vannak, így ha új nyelvekkel bővül az API, ezt a map-et is szükség szerű frissíteni.

## ForditoService

Ez a Service is a Yandex API használatán alapul. Legjelentősebb funkcióját a *createForditas(szo : Forditas)* függvény valósítja, mely a paraméterben kapott objektumban ( melyek mezői egy-egy Nyelv típusú forrásnyelv és célnyelv, valamint egy string típusú szó) szereplő szónak egy API kéréssel lekéri a szinonimáit/fordításait, valamint az ehhez kapcsolódó információkat (pl. szófaj) és az egyes fordításokat egy Szo típusú objektummal reprezentálja. Mivel az API angol nyelven jeleníti meg a szófajokat, ezért a NyelvService-nél is látott map-es módszert használjuk arra, hogy az alkalmazás során magyarul jelenjen meg a szófaj.

## Yandex Dictionary API

A Yandex Dictionary API-jának funkciója, az, hogy egy adott szóhoz, megadva a forrás- és célnyelvet, eredményként megkapjuk a lehetséges fordításokat és az ehhez kapcsolatos információkat (pl. szófaj). Ahhoz, hogy ezt az API-t használni tudjunk egy kulcsra (API Key) van szükségünk, melyet ingyenesen e-mail cím megadásával tudunk igényelni a yandex oldalán. Az API 2 metódussal rendelkezik : *getLangs*, valamint *lookup*, a szótár készítése során mind a két metódust igénybe vettük. Az adatokat JSON formátumban várjuk, így a JSON alapú request-et vettük igénybe.

A *getLangs* metódus az alábbi URL-en érhető el:

<https://dictionary.yandex.net/api/v1/dicservice.json/getLangs?key=APIkey>, amelyben az APIkey helyére a saját API kulcsunkat kell beírjuk. Ezen metódus az alábbi eredményt adja vissza, melyet a fent leírt megoldások segítségével könnyebben kezelhetővé tudjuk tenni.

```
[ "be-be", "be-ru", "bg-ru", "cs-cs", "cs-en", "cs-ru", "da-en", "da-ru", "de-de", "de-en", "de-ru", "de-tr", "el-en", "el-ru", "en-cs", "en-da", "en-de", "en-el", "en-en", "en-es", "en-et", "en-fi", "en-fr", "en-it", "en-lt", "en-lv", "en-nl", "en-no", "en-pt", "en-ru", "en-sk", "en-sv", "en-tr", "en-uk", "es-en", "es-es", "es-ru", "et-en", "et-ru", "fi-en", "fi-ru", "fi-fi", "fr-fr", "fr-en", "fr-ru", "hu-hu", "hu-ru", "it-en", "it-it", "it-ru", "lt-en", "lt-lt", "lt-ru", "lv-en", "lv-ru", "mhr-ru", "mrj-ru", "nl-en", "nl-ru", "no-en", "no-ru", "pl-ru", "pt-en", "pt-ru", "ru-be", "ru-bg", "ru-cs", "ru-da", "ru-de", "ru-el", "ru-en", "ru-es", "ru-et", "ru-fi", "ru-fr", "ru-hu", "ru-it", "ru-lt", "ru-lv", "ru-mhr", "ru-mrj", "ru-nl", "ru-no", "ru-pl", "ru-pt", "ru-ru", "ru-sk", "ru-sv", "ru-tr", "ru-tt", "ru-uk", "ru-zh", "sk-en", "sk-ru", "sv-en", "sv-ru", "tr-de", "tr-en", "tr-ru", "tt-ru", "uk-en", "uk-ru", "uk-uk", "zh-ru" ]
```

A *lookup* metódus az alábbi URL-en érhető el:

<https://dictionary.yandex.net/api/v1/dicservice.json/lookup?key=APIkey&lang=hu-ru&text=vár> . Az APIkey helyére ahogy az előbbi esetben, most is a saját Yandex API-s kulcsunkat kell beírjuk, a lang= után a nyelvek rövidítéseivel meg adhatjuk, hogy mi a forrás- és célnyelv (pl. jelen esetben magyarról fordít

oroszra), és a text parameter után, hogy milyen szót fordítsunk le, amely jelen esetben a “vár” szó. A válaszul kapott JSON file az alábbi formátumú.

```
{ "head": {}, "def": [ { "text": "vár", "pos": "noun", "tr": [ { "text": "замок", "pos": "noun", "gen": "м", "syn": [ { "text": "крепость", "pos": "noun", "gen": "ж" }, { "text": "дворец", "pos": "noun", "gen": "м" } ] }, { "text": "erőd" } ] }, { "text": "vár", "pos": "verb", "tr": [ { "text": "подождать", "pos": "verb", "asp": "сов" } ] } ] }
```

Az visszaadott adatok jelentései a következők:

- “def” : ezen belül található minden lényeges adat, egy def-en belüli objektum a szó egy-egy értelmezését reprezentálja
- “text” : a szó leírt formában
- “pos” : szófaj
- “tr” : fordítások, melyeken belül (text, pos) formában megtalálhatók a szavak ( néhány nyelv esetében a text, és pos mellett egy gen tulajdonság is megjelenik, amely a szó nemére utal, ám ezt az információt nem dolgozzuk fel a működés során)
- “mean” : a fordítandó szó szinonimái

## Fordítás

Az alábbiakban áttekintjük, hogy mi történik onnantól kezdve, hogy a felhasználó megnyomja a “Fordít” gombot, addig, amíg az eredmények megjelennek a felületen.

1. A kattintás során a ForditoService *fordit()* függvénye hívódik meg.

```
2. <button type="button" class="btn btn-dark" (click)="fordit()" [disabled]="somethingEmpty()">
3.   Fordít
4. </button>
```

2. A fordit() függvény meghívja a forditoService createForditas() függvényét és átadja a forditandoSzo paramétert, mely egy Forditas típusú objektum, így megtalálható benne, a forrás-, a célnyelv és az aktuális szó is, majd ezután a Service által lekéri az eredményt, és ezt Observable típusú alakítja, így a HTML-ből is el tudjuk érni.

```
1. this.forditoService.createForditas(this.forditandoSzo);
2. var _forditasok = this.forditoService.getForditasok();
3. this.forditasok = of(_forditasok);
```

3. Első lépésben a függvény inicializálja az eredményeket (this.forditas) üres tömbre, majd véghez hajtja az API hívást, melynek visszatérési értéke egy Observable Javascript Object. A szo nevű objektum a paraméterben érkezett Forditas típusú objektum.

```
1. this.forditasok = [];
```

```

2. var forrasRovidites = szo.forrasNyelv.rovidites;
3. var celRovidites = szo.celNyelv.rovidites;
4. var URL = "https://dictionary.yandex.net/api/v1/dicservice.json/lookup?key=" +
5.   this.APIKey +
6.   "&lang=" +
7.   forrasRovidites + "-" + celRovidites +
8.   "&text=" + szo.szo;
9. this.forditasObs = this.http.get<ResponseJson>(URL);

```

4. Az API kérés megtörténésével az Observable típus subscribe függvény segítségével, létrehozza a Szo osztályú eredményeinket, melyek olvashatóbbak és könnyebben kezelhetőek. Minden egyes értelmezés (def), minden egyes fordításhoz (tr) asszociálunk egy Szo objektumot, melyben megtalálható maga a szó, a nyelve, valamint a szófaja.

```

1. this.forditasObs.subscribe(data => {
2.   var id = 0;
3.   for(let definition of data.def) {
4.     for(let translation of definition.tr) {
5.       var szofajMagyar = this.szofajAngolMagyar.get(translation.pos);
6.       var leforditottSzo = {id : id, nyelv: szo.celNyelv.nev, text : translation.text,
7.         szofaj :szofajMagyar}
8.       id++;
9.       this.forditasok.push(leforditottSzo);
10.    }
11.  }
12. });

```

5. Miután az componens lekérdezte a fordításokat a service-től a HTML-be betöltődnek az eredmények, és ekkor már a user is láthatja az eredményt.

```

1. <li *ngFor="let forditas of (forditasok | async)" (click)="wordWithDetails = forditas">
2.   <h6 class="forditas">{{forditas.text}}</h6>
3. </li>

```