

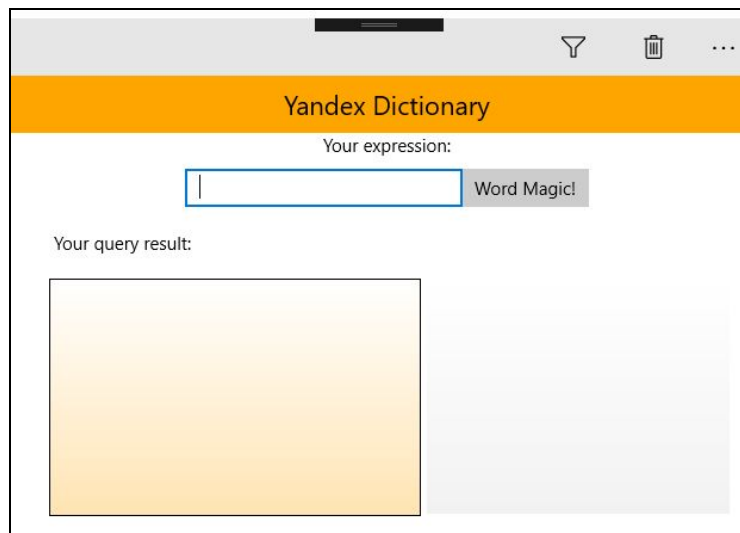
Yandex & Thesa szótár

Kliensoldali technológiák házi feladat - Készítette: Paréj Balázs (E1512Q)

Megjegyzés: A program során a template10 keretrendszert használtam, ami új projektben valamilyen oknál fogva nem működött. Ezért a gyakorlat során használt Cookbook projektet használtam, tehát előfordulhat, hogy néhány helyen maradt erre utaló jel.

Összefoglaló

A Yandex & Thesa szótár alkalmazás lehetővé teszi azt, hogy a felhasználó valamely nyelvből egy másikba fordítson egy-egy kifejezést, illetve segíthet kibővíteni szókincsünket, hiszen rengeteg nyelvhez tartozik szinonima szótár is. A program elindítása történhet futtatható fájl (.exe) megnyitásával, illetve Visual Studioban is elindíthatjuk.



Indítás után a fenti kép fogad bennünket. Értelmszerűen, a “Your expression” felirat alatti szövegdobozba vihetjük be a feldolgozandó szöveget. A CommandBar (szürke sáv felül) második gombjának ikonja egy szűrő, itt egy lenyíló menüben be tudjuk állítani a kimeneti, bemeneti nyelvet, illetve, hogy szinonimát vagy fordítást szeretnénk.

Ha olyan kifejezést adunk meg, amihez talált egyezést (azaz benne van a szótárban), akkor az a narancssárga átmenetes boxban lesz látható, illetve a jobb oldali lista nézetben egy rövid keresési előzmény látható, ami a CommandBar Kuka ikonjára kattintva üríthető.

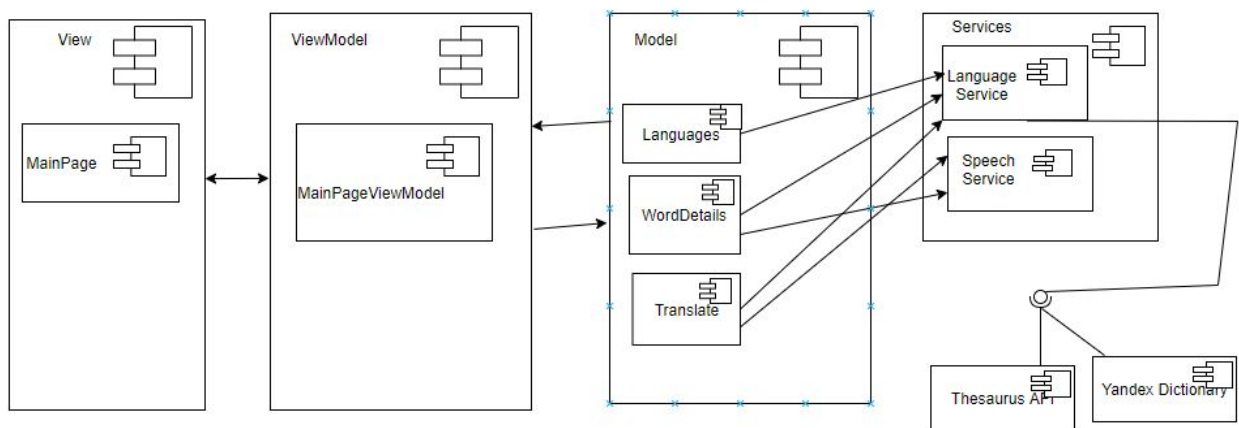
Architektúra

Az alkalmazást UWP-vel valósítottam meg, a fordításhoz a Yandex Dictionary API-t használtam, a szinonimák kereséséhez pedig a Thesaurus API-t használtam.

A rendszer felépítése az MVVM architektúrát követi, a View és ViewModel között Databinding, a ViewModel és Model között pedig "kölcsonös" módosítások.

Az MVVM megvalósításához a template10 keretrendszert használtam.

Az ábra a diagrams.net-tel készült.



View:

- **MainPage:** A program designját tervezve az egyszerűsége törekedtem, ezáltal a legegyszerűbb megvalósítás az volt, hogy egy nézet van.

ViewModel:

- **MainPageViewModel:** A View-ból kifolyólag egy viewhoz csak egy ViewModel tartozik, ez kapcsolja össze a modelleket és a nézeteket. Értesíti a nézetet ha megváltozott a modelpéldány tartalma.

Model:

- **Languages:** A Yandex API által lekérdezhető nyelveket tároló model. JSON alapján generált.
- **WordDetails:** A Thesaurus API szinonima lekérdező válasza alapján generált osztály.
- **Translate:** A Yandex API fordítás válaszából generált osztály.

Services:

- **LanguageService:** A hálózati kommunikáció, API hívások helye.
- **SpeechService:** Szövegfelolvasás.

Osztályok bemutatása

Megjegyzés: Az előző ábrából kimaradtak a Converter osztályok, mert nem tudtam besorolni őket.

Converter osztályok: IValueConverter interfésszel

- ModeToBooleanConverter: Ez az osztály a nevéből sugallóan az alkalmazás működési módját alakítja át "True/False"-ra. Arra használok, hogy azokat az elemek Visibility attribútumát állítsam, amelyek nem kellenek mindkét működési módhoz (fordítás és szinonimák). A Visibility attribútum azonban nem true vagy false értéket vár, hanem Visible illetve Collapsed értékeket, tehát ezeket adja vissza.
- BooleanToVisibilityConverter: Ez az előzőhöz hasonló, viszont ez a nyelvek alapján rejt el az elemeket. Főként arra van használva, hogy a hanglejátszó gomb csak a támogatott nyelvek esetén jelenjen meg.

LanguageService:

Itt a függvények aszinkron módon kommunikálnak a szerverekkel, ezzel biztosítva, hogy a GUI ne akadjon be. Van egy "alap" getAsync metódus, ez egy HTTP klienssel lekéri JSON adatformátumban melyet majd a publikus lekérdező függvények fognak a megfelelő osztályba átkonvertálni.

SpeechService:

Itt történik a beszéd szintetizálás, azaz a bemeneti szöveget képes audióvá alakítani. Ehhez a WINRT SpeechSynthesizerét használtam.

Érdemes megemlíteni, hogy ez csak akkor működik, ha van a Windows rendszerünkön alapból beszédhang felolvasáshoz. Ha nincsen, az Ország & Régió beállításban tudunk letölteni, és ezt beállítani alapértelmezettnek.

Jelen esetben úgy működik, hogy a beszédhang adatbázisban keresünk egy angol hangot, ha nincs, akkor a rendszerünk alapértelmezett hangja fog megszólalni. Ez magyar Windows esetén Microsoft Szólabólcsot jelent.

A többi osztályt röviden lásd fent.

Szerverrel történő kommunikáció:

Kommunikáció a Yandex-szel:

A Yandex egy orosz multinacionális vállalat, amely internetes termékekre és szolgáltatásokra specializálódott. Az ő termékük az úgynevezett Yandex Dictionary API amellyel HTTP get requestekkel kommunikálhatunk.

Ehhez a termékhez APIkey kötelezően generálandó (anélkül "Denied Request"-et kapunk), amit ingyen megtehetünk - feltéve, hogy nincs meglehetősen nagy forgalmunk.

Az első GET request amit megvalósítottam, a getLangs metódus volt. A honlapon lehet választani milyen interfésszel szeretnénk dolgozni, XML, JSON illetve JSONP a választék.

Én a JSON interfésszel dolgoztam, amit a következő URL-lel lehet meghívni:

<https://dictionary.yandex.net/api/v1/dicservice.json/getLangs?key=APIkey>

ahol az APIkey a saját kulcsunk.

A hívás eredménye (részlet):

```
["ru-ru", "ru-en", "ru-pl", "ru-uk", "ru-de", "ru-fr", "ru-es", "ru-it", "ru-tr", "en-ru", "en-en", "en-de", "en-fr", "en-es", "en-it", "en-tr", "pl-ru", "uk-ru", "de-ru", "de-en", "fr-ru", "fr-en", "es-ru", "es-en", "it-ru", "it-en", "tr-ru", "tr-en"]
```

Egyébként nem meglepő módon, eléggé szláv orientált adatbázisról van szó, de használható korlátozottan például magyarra is.

A másik HTTP get maga a szótári lekérdezés, aminek a neve lookUp.

Az előzőhöz hasonlóan, itt is 3 interfészből választhatunk, a JSON meghívásához a következő URL tartozik:

<https://dictionary.yandex.net/api/v1/dicservice.json/lookup?key=APIkey&lang=en-ru&text=time>

ahol az APIkey a kulcsunk, az en-ru a kiválasztott nyelv, a time pedig a fordítandó szó. Ezen kívül be lehet állítani pl. flageket is, amellyel például "családi" szűrőt rakhatunk a keresésünkre.

Thesaurus API

A Thesaurus a szinonimák lekérdezéséhez használatos, ez szintén HTTP get requesttel kommunikál, a következő url-lel érhetjük el:

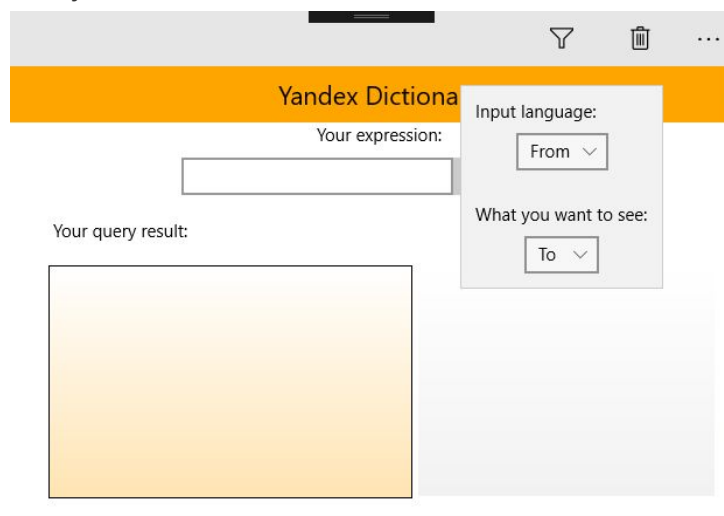
http://thesaurus.altervista.org/thesaurus/v1?word=peace&language=en_US&key=test_only&output=json

ahol a word a keresett szó, a language a bemeneti és kimeneti nyelv, ami ugyanaz, key az api kulcs, output alapértelmezetten XML de én JSON-nel dolgoztam, a konzisztencia kedvéért.

API kulcs az előzőhöz hasonlóan ingyen igényelhető, de bizonyos hívás felett fizetni kell érte.

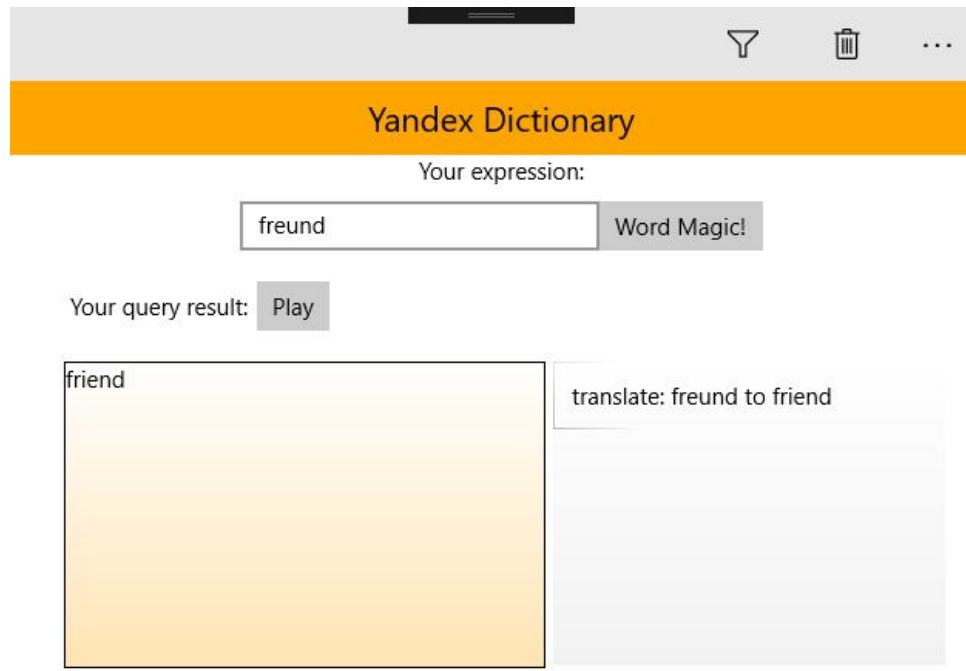
Fordítás hívásának menete

- 1) A program elindítása.
- 2) Rákattintunk a szűrő jelre



- 3) A "what you want to see" alatt válasszuk ki a translatet, majd kiválasztjuk az Input és az Output language-t is. Fontos, hogy először az inputot hívjuk meg, mert az alapján jelennek meg az output nyelvek. Esetünkben legyen en-de.
- 4) Ezután beírhatjuk a kívánt szót, és nyomjunk a szövegdoboz melletti gombra. Esetünkben legyen Freund.
- 5) API hívása a gomb lenyomásával:
 - a) A következő url generálódik:
["https://dictionary.yandex.net/api/v1/dicservice.json/lookup?key=dict.1.1.20200505T110543Z.b9fabeb7eb7fc41c.d5a0850505ea646a3fa3a0bd582ef36071a7f7fa&lang=de-en&text=freund"](https://dictionary.yandex.net/api/v1/dicservice.json/lookup?key=dict.1.1.20200505T110543Z.b9fabeb7eb7fc41c.d5a0850505ea646a3fa3a0bd582ef36071a7f7fa&lang=de-en&text=freund)

- b) Válaszként ezt a stringet kapjuk json formában:
- ```
{"head":{}, "def": [{"text": "Freund", "pos": "noun", "gen": "m", "ts": "fʁɔ̃d nt", "fl": "-s, -es; -e", "tr": [{"text": "friend", "pos": "noun", "syn": [{"text": "boyfriend", "pos": "noun"}], "mean": [{"text": "Freundin"}, {"text": "Boyfriend"}]}]}
```
- c) Ezt deszerializáljuk a megfelelő objektumnak.
- d) Visszatérés az objektummal.
- 7) Az objektum értékének átadása egy stringnek, ami bindolva van a nézeten lévő textblockhoz.



- 8) Nyomjunk rá a Play gombra, hogy meghallgassuk Microsoft Mike selymes hangját.