# RSS feeds to facebook page

Tento projekt je webová aplikace napsaná v **Node.js**, která spravuje uživatele, získává jejich přístupové tokeny na Facebook a automaticky publikuje články z RSS feedů na jejich Facebookové stránky. Aplikace také kontroluje expiraci přístupových tokenů pomocí cron úloh a umožňuje uživatelům reautorizovat své účty.

## Knihovny, které byly použity

* **Express.js** - Webový framework pro Node.js [*npm install express]*
* **MySQL** - Pro práci s MySQL databází [*npm install mysql*]
* **Body-parser** - Pro parsování JSON a formulářových dat v požadavcích [*npm install body-parser*]
* **Path** - Pro práci s cestami k souborům a adresáři [*mělo by být součástí Node.js*]
* **Node-cron** - Knihovna pro plánování periodických úloh (cron jobs) [*npm install node-cron*]
* **Axios** - HTTP klient pro odesílání požadavků, používá se pro volání API a stahování obrázků [*npm install axios*]
* **Dotenv** - Pro práci s proměnnými ze souboru .env [*npm install dotenv*]
* **RSS-parser** - Pro parsování RSS feedů [*npm install rss-parser*]
* **FormData** - Pro odesílání formulářových dat, používá se při nahrávání obrázků na Facebook [*npm install form-data*]
* **Cheerio** - Pro manipulaci s HTML a extrakci dat z HTML struktury [*npm install cheerio*]

Souhrnný příkaz: [*npm install express mysql body-parser node-cron axios dotenv rss-parser form-data cheerio*]

## Konfigurace prostředí

Soubor .env pro nastavení citlivých informací, jako jsou přístupové údaje k databázi nebo k Facebook API.

Vzor obsahu .env souboru:

DB\_HOST=localhost  
DB\_USER=root  
DB\_PASSWORD=heslo  
DB\_NAME=databaze  
APP\_ID=fb\_app\_id  
APP\_SECRET=fb\_app\_secret

Moje nastavení:

DB\_HOST=localhost  
DB\_USER=root  
DB\_PASSWORD=  
DB\_NAME=db\_facebook\_users  
  
APP\_ID=2289710858046148  
APP\_SECRET=e5d6e753a3d63e147746ef5f96ad130b

## Struktura

**server.js** - hlavní skript, který obsahuje veškerý backend

**\public** - obsahuje html pro prvotní přihlášení uživatele (index.html) a html pro reautorizaci uživatele (reautorizace.html)

# Funkce v server.js

### Endpoint /save-user-data

Přijímá URL RSS feedu, Facebook User ID a short user access token z html stránky index.html.

Následně volá funkce getLongLivedToken a getPageAccessToken pro získání long user access tokenu, který je následně převeden na page access token.

Uživatele uloží/aktualizuje do databáze.

Endpoint je volán přes index.html.

### Endpoint /update-tokens

Aktualizuje access tokeny, pokud je potřeba reautorizace uživatele.

Endpoint je volán přes reautorizace.html

### getLongLivedToken

Funkce získává long-lived user access token pomocí krátkodobého access tokenu.

Facebook Graph API: https://graph.facebook.com/v12.0/oauth/access\_token?grant\_type=fb\_exchange\_token&client\_id=${appId}&client\_secret=${appSecret}&fb\_exchange\_token=${shortUserAccessToken}

### getPageAccessToken

Funkce získává page access token a ID Facebookové stránky pomocí long-live user access tokenu, zároveň získává hodnotu data\_access\_expires\_at.

Facebook Graph API:

Získání hodnoty hodnotu data\_access\_expires\_at:

https://graph.facebook.com/debug\_token?input\_token=${longLivedUserToken}&access\_token=${appId}|${appSecret}

Získání page access tokenu:

https://graph.facebook.com/v12.0/me/accounts?access\_token=${longLivedUserToken}

data\_access\_expires\_at se ukládá ve formánu unix timestampu, pro případné přečtení je potřeba ho převést na čitelné datum (nebo využít <https://www.unixtimestamp.com/>)

### uploadImageToFacebook

Funkce pro nahrávání obrázků na Facebook stránku.

Obrázek je nejprve stažen pomocí axios, a poté nahrán na Facebook pomocí FromData.

Obrázek je nahrán jako Facebook fotka (${pageId}/photos), ale díky published – false se obrázek nepublikuje, pouze se „uloží“.

### postToFacebookWithImage

Slouží k publikaci příspěvku s obrázkem.

### postToFacebook

Slouží pro publikaci textového příspěvku bez obrázku.

### processRSSFeed

Hlavním úkolem této funkce je zpracovat RSS feed z url adresy uložené v databázi. Zpracovává každý článek v RSS, kontroluj, zda již byl publikován, a případně ho publikuje.

**Tady nevím, zda je možné to zpracovat bez databáze, časem se může dost zaplňovat.**

### constructImageUrlFromContent

Pokud uživatel přidá obrázek do těla aktuality (pole Text) extrahuje hodnoty nid a oid z HTML obsahu články, aby vytvořila URL pro obrázek, který bude použit pro publikaci na Facebook. Pokud uživatel chce publikovat obrázek z těla aktuality, nesmí být obrázek vložen v úvodním obrázku, pokud ano vloží se obrázek z úvodního obrázku.

### cleanHTMLContent

Při publikaci těla aktuality (pole Text) se vkládání i HTML elementy, pomocí této funkce se jich zbavíme a nahráváme jen text těchto elementů, obrázky jsou vyloučeny. Odkazy (a elementy) se automaticky převádí na hypertextové odkazy v textu.

Funkce se zároveň zbavuje kódových mezer (&nbsp) apod.

### formatDateTime

Formátuje aktuální datum a čas do formátu YYYY-MM-DD HH:MM:SS, jedná se pouze o zbytečnou funkci. Pouze, aby console logy měli čas spuštění.

### extractDomainFromRSSUrl

Opět zbytečná funkce, je zde pouze pro lepší identifikaci uživatele, pro kterého má vypršet access token, respektive data\_access (3 měsíce).

Lze využít, ale pro generování odkazu pro autorizaci – např.

http://${domain}/reautorizace

### checkDataAccessExpiration

Tato funkce kontroluje hodnotu data\_access\_expires\_at, která je uložena v databázi.

Pokud je expirace blízko, respektive 10 dní před expiraci (nastaveno v daysBeforeExpiration), může poslat uživateli notifikaci pro autorizaci. Lze využít knihovnu *nodemailer*, kde lze nastavit mailový server a odeslat zprávu na zvolený email. Email uživatele lze získat při prvotním přihlášení k facebook propojení, v tomto řešení není zatím implementováno.

V poslední podmínce lze pro testování nahradit timeBeforeExpiration za statickou hodnotu, např. pokud je data\_access\_expires\_at 1737070198 (lze získat z databáze), lze vložit větší hodnotu 1737070199 pro vyvolání reautorizace.

### Cron Jobs

1. Kontroluje expiraci data\_access\_expires\_at, zatím nastaveno na každých 20 vteřin pro testování, poté se může spouštět každých ??? (**potřeba domyslet, aby se vše pokrylo**)
2. Zpracovává RSS feedy, opět kvůli testování nastaveno na 20 vteřin. Může se poté spouštět každou 1. nebo 2. hodinu.

## Možnosti publikace aktualit a jejich finální propsání na Facebookové stránky

**Pokud chce uživatel ukládat na Facebook obsah z aktualit (pole Text) nesmí vkládat Úvodní text => Úvodní text a Text mají v RSS feedu stejnou značku (description) a pokud je vložen úvodní text, vždy ji prioritně přebírá.**

1. **možnost**

Obsah obrázku text, snímek obrazovky, Písmo, bílé

Popis byl vytvořen automaticky

Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Výsledek na FB: Publikuje se Titulek a Text

Obsah obrázku text, snímek obrazovky, Písmo, bílé

Popis byl vytvořen automaticky

1. **možnost**

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

Popis byl vytvořen automaticky Obsah obrázku nábytek, pohovka, Gauč, Poduška

Popis byl vytvořen automaticky

Obsah obrázku text, Písmo, snímek obrazovky, bílé

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Popis byl vytvořen automaticky

Výsledek na FB: Publikuje se Titulek, Úvodní obrázek a Text

Obsah obrázku nábytek, pohovka, Gauč, gauč

Popis byl vytvořen automaticky

1. **možnost**

**Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky**

**Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky**

**Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky**

**Obsah obrázku pohovka, nábytek, Gauč, gauč

Popis byl vytvořen automaticky**

Výsledek na FB: Publikuje se Titule, Text a obrázek z textu

**Obsah obrázku nábytek, pohovka, gauč, Gauč

Popis byl vytvořen automaticky**

1. **možnost**

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky Obsah obrázku nábytek, pohovka, Gauč, gauč

Popis byl vytvořen automaticky

Obsah obrázku nábytek, židle

Popis byl vytvořen automaticky

Výsledek na FB: Publikuje se Titulek, Úvodní obrázek, Text, **obrázek z textu se nepublikuje, úvodní obrázek má prioritu**

Obsah obrázku nábytek, pohovka, Gauč, Poduška

Popis byl vytvořen automaticky

1. **možnost**

**Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky**

**Obsah obrázku text, snímek obrazovky, Písmo, řada/pruh

Popis byl vytvořen automaticky**

**Obsah obrázku nábytek, pohovka, gauč, Gauč

Popis byl vytvořen automaticky**

Výsledek na FB: Publikuje se Titulek, Text a 1. obrázek z textu

Obsah obrázku nábytek, pohovka, Gauč, gauč

Popis byl vytvořen automaticky