

TOTEMAPP

GitHub: <https://github.com/ScoutsGidsenVL/Totemapp>

Bij vragen of onduidelijkheden:
Frederick Eskens (frederickeskens@gmail.com)

Aanpassingen maken

1. Databank

Als er totems, eigenschappen of tips aangepast moeten worden, dient dit in de databank te gebeuren.

1.1. Kleine aanpassingen

Indien er slechts een kleine aanpassing gedaan moet worden zoals een schrijffout aanpassen of een synoniem toevoegen of verwijderen, is het minder werk om deze twee keer aan te passen (in de algemene databank en in de app-databank) dan deze alleen aan te passen in de algemene databank en vervolgens te exporteren.

Open 'totems.sqlite' in de map TotemAppCore\SharedAssets. Dit kan met *DB Browser for SQLite* (sqlitebrowser.org). Maak vervolgens de aanpassingen en sla de databank op. Dit is ook de manier om tips aan te passen/te verwijderen/toe te voegen.

1.2. Grote aanpassingen

Indien er grotere aanpassingen moeten gebeuren zoals een update van de totems, is het best om de gewenste info te exporteren uit de algemene databank. Exporteer de gewenste info met Navicat aan de hand van de export wizard (exporteer naar .sql-bestand). Dit bestand bevat alle nodige insert statements. De insert into's zijn leeg. Gebruik een text editor om de juiste tabel als doel in te vullen. Enkele handige queries:

- Exporteren van de totems

```
SELECT node_revisions.nid, node_revisions.title, node_revisions.body,
syn.field_totem_synoniemen_value AS synonyms
FROM node_revisions INNER JOIN (SELECT nid, MAX(vid) AS maxVid FROM
node_revisions GROUP BY nid) gr ON node_revisions.nid = gr.nid AND
node_revisions.vid = gr.maxVid
INNER JOIN (SELECT content_type_totem.nid,
content_type_totem.field_totem_synoniemen_value
FROM content_type_totem INNER JOIN (SELECT nid,
MAX(vid) AS maxVid FROM content_type_totem GROUP BY nid) gr ON
content_type_totem.nid = gr.nid AND content_type_totem.vid = gr.maxVid
ORDER BY content_type_totem.nid) syn ON node_revisions.nid = syn.nid
WHERE node_revisions.nid IN (SELECT nid FROM node WHERE type='Totem')
ORDER BY title
INSERT INTO('totem')
```

De totem tabel heeft een extra kolom *number*. Deze kolom houdt automatisch de nummers van de totems bij. Aangezien deze kolom niet bestaat in de algemene databank, moet deze eerst verwijderd worden opdat de data uit het .sql-bestand correct geïmporteerd kan worden. Nadien kan de kolom terug toegevoegd worden (``number` INTEGER PRIMARY KEY AUTOINCREMENT`). Dit kan ook aan de hand van *DB Browser for SQLite*.

- Exporteren van de eigenschappen

```
SELECT tid, name
FROM term_data
WHERE term_data.tid IN (SELECT term_node.tid
                        FROM term_node INNER JOIN (SELECT nid, MAX(vid) AS
maxVid FROM term_node GROUP BY nid) gr ON term_node.nid = gr.nid AND
term_node.vid = gr.maxVid
                        WHERE term_node.nid IN (SELECT nid FROM node WHERE
type='Totem'))
INSERT INTO('eigenschap')
```

- Exporteren van de tussentabel

```
SELECT term_node.tid, term_node.nid
FROM term_node INNER JOIN (SELECT nid, MAX(vid) AS maxVid FROM term_node
GROUP BY nid) gr ON term_node.nid = gr.nid AND term_node.vid = gr.maxVid
WHERE term_node.nid IN (SELECT nid FROM node WHERE type='Totem')
INSERT INTO('totem_eigenschap')
```



Wanneer de databank is aangepast, moet de databank versie geïncrementeerd worden. Dit moet gebeuren in 'Database.cs' in de map TotemAppCore\SGV\DL\. **Daar moet const int DATABASE_VERSION met één verhoogd worden.** Indien dit niet gebeurt, zal de gebruiker nog steeds de oude versie van de databank gebruiken!

2. Startscherm

2.1. Android

- **Titel**

Ga naar TotemAndroid\Resources\layout\Main.xml. Verander het veld `android:text="TOTEMAPP"` van de CustomFontTextView met id `totemapp_title`.

- **Knoppen**

Vergelijkbaar met titel.

- **Tips**

De tips staan in de databank. Zie databank aanpassen.

- **Afbeeldingen**

De kleur van het gras kan aangepast worden door `android:color="#6c773b"` aan te passen in het bestand TotemAndroid\Resources\drawable\grass_green.xml.

De afbeeldingen kunnen aangepast worden door de bestanden 'Berg.png' en 'Totem.png' aan te passen in de map TotemAndroid\Resources\drawable\. Naargelang de afbeelding zal deze verplaatst moeten worden. Doe dit door de waardes van `marginBottom/marginLeft/marginRight` aan te passen.

- **Disclaimer onderaan**

De tekst kan aangepast worden door `android:text="door..."` aan te passen in `TotemAndroid\Resources\layout\InfoToast.xml`.

2.2. iOS

- **Titel**

Ga naar `TotemApplos\SGV\TotemApp\Controllers\Main\MainViewController.cs`. Daar kan je in de methode `setData()` de tekst aanpassen door `lblTitle.Text` te veranderen.

- **Knoppen**

Vergelijkbaar met titel.

- **Tips**

De tips staan in de databank. Zie databank aanpassen.

- **Afbeeldingen**

De kleur van het gras en de plaatsing van de afbeeldingen kunnen aangepast worden door `TotemApplos\SGV\TotemApp\Controllers\Main\MainViewController.xib` te openen met de Xcode Interface Builder. De afbeeldingen zelf kunnen veranderd worden in de `setData()` methode van `TotemApplos\SGV\TotemApp\Controllers\Main\MainViewController.cs`.

- **Disclaimer onderaan**

Vergelijkbaar met titel.

3. Titels

3.1. Android

Bij elk scherm hoort een Activity en deze Activities staan in `TotemAndroid\SGV\`. Om een titel aan te passen van een scherm, moet in de `OnCreate()` methode van de bijhorende Activity `title.Text = "<tekst>"` worden veranderd. De Activities hebben logische namen (behalve het scherm van individuele weergave, die heet 'TinderActivity.cs'). 'BaseActivity.cs' is een abstracte superklasse en moet niet aangepast worden.

3.2. iOS

Bij elk scherm hoort een ViewController en deze ViewControllers staan in `TotemApplos\SGV\Totemapp\Controllers`. Om een titel aan te passen van een scherm, moet in de `setData()` methode van de bijhorende ViewController `lblTitle.Text = "<tekst>"` worden veranderd. De ViewControllers hebben logische namen (behalve het scherm van individuele weergave, die heet 'TinderEigenschappenViewController.cs'). 'BaseViewController.cs' is een abstracte superklasse en moet niet aangepast worden.

4. Totemisatie checklist

4.1. Android

- **Checklist**

De checklist kan aangepast worden in het bestand `TotemAndroid\Resources\values\arrays.xml`. Daar staat per puntje van de checklist een string-array van items. Items worden altijd voorgegaan door hun type gevolgd door een liggend streepje (bvb. `n_welke_totemvorm?`). `n` staat voor *normal*, `i` voor *indented* en `h` voor *head*. Om een item aan een puntje toe te voegen, volstaat het om dit bij in de 'arrays.xml' te

zetten. Wanneer een nieuw puntje toegevoegd zou worden, moet er nog een extra aanpassing gebeuren. In TotemAndroid\SGV\TotemisatieChecklistActivity.cs moet dit puntje toegevoegd worden in de methode FillDictGroup(), analoog aan degene dat er al staan.

- **Header/footer**

Om de header of de footer aan te passen moet de overeenkomstige string in TotemAndroid\Resources\values\Strings.xml worden veranderd.

4.2. iOS

- **Checklist**

De checklist kan aangepast worden in het bestand TotemApplos\SharedAssets\checklist.xml. Daar staat per puntje van de checklist een string-array van items. De naam van elk puntje staat als name parameter bij de string-array. Items worden altijd voorgegaan door hun type gevolgd door een liggend streepje (bvb. n_welke_totemvorm?). n staat voor *normal*, i voor *indented* en h voor *head*.

- **Header/footer**

De header en footer kunnen aangepast worden in de setData() methode van 'ChecklistViewController.cs' in TotemApplos\SGV\Totemapp\Controllers\ door respectievelijk lblHead.Text en lblFoot.Text te wijzigen.

5. Logo

5.1. Android

Om de logo's voor Android te genereren werd volgende tool gebruikt: gieson.com/Library/projects/utilities/icon_slayer. Alle instellingen aan de rechterkant moeten afstaan, behalve *Rounded Corners*. Nadat de logo's geëxporteerd zijn, moeten de juiste logo's in de juist mappen gezet worden. Importeer vanuit de map *sizes* de logo's naar de correcte map in TotemAndroid\Resources met Visual Studio. Klik rechtermuisknop op de map en dan *Add > Existing Item...* Kies telkens het correcte bestand voor de map aan de hand van onderstaande tabel.

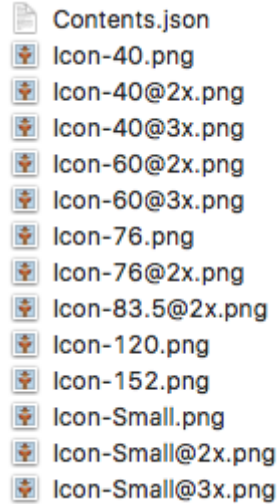
Map	Resolutie
drawable-hdpi	72 x 72
drawable-mdpi	48 x 48
drawable-xhdpi	96 x 96
drawable-xxhdpi	144 x 144
drawable-xxxhdpi	192 x 192

Indien de gewenste resolutie niet in de map *sizes* staat, kunnen ze door de tool gegenereerd worden door rechts onderaan *Custom* in te vullen. De nieuwe logo's moeten de naam 'icon.png' krijgen in alle mappen (verwijder of hernoem het oude logo). In de code moet verder niets aangepast worden.

Ten slotte moet op de Google Developer Console bij *Winkelvermelding* het logo ook aangepast worden. Hier is een resolutie van 512 x 512 vereist.

5.2. iOS

Om de logo's voor iOS te genereren werd volgende tool gebruikt: makeappicon.com. Geef een e-mailadres in en download de gemaakte zip. In de map *ios* van het .zip-bestand staat een map *AppIcon.appiconset*. Dupliceer in deze map de bestanden 'Icon-60@2x.png' en 'Icon-76@2x.png'. Hernoem ze respectievelijk 'Icon-120.png' en 'Icon-152.png'. De map *AppIcon.appiconset* ziet er nu zo uit:



Verwijder in Xamarin Studio de huidige map *AppIcon.appiconset* die staat in *TotemApplos\Resources\Images.xcassets*. Voeg de nieuwe map vervolgens toe door rechtermuisknop te klikken op *Images.xcassets*. Doe dan *Add > Add Existing Folder...* Kies vervolgens de nieuwe map *AppIcon.appiconset* en voeg hem (en al zijn inhoud) toe.

Ten slotte moet op iTunes Connect bij *General App Information* (bij de build onder *iOS app* aan de linkerkant) het logo ook aangepast worden. Hier is een resolutie van 1024 x 1024 vereist.