







YT Creator Tools

Outils professionnels pour créateurs YouTube avec extraction de sous-titres, métadonnées, et téléchargement de miniatures.

Fonctionnalités

-  Extraction de métadonnées vidéos YouTube
-  Téléchargement de miniatures en différentes qualités
-  Extraction de sous-titres avec `youtube-transcript-api`
-  Authentification Firebase (email + Google)
-  Interface PWA responsive
-  API REST avec Flask

Structure du projet

```
yt-creator-tools/  
├── app.py ..... # Serveur Flask principal  
├── requirements.txt ..... # Dépendances Python  
├── gunicorn_config.py ..... # Configuration Gunicorn  
├── render.yaml ..... # Configuration Render  
├── .env.example ..... # Variables d'environnement  
├── README.md ..... # Ce fichier  
├── static/  
│   ├── index.html ..... # Interface utilisateur  
│   ├── manifest.json ..... # Manifeste PWA  
│   ├── service-worker.js # Service Worker  
│   └── privacy_policy.html # Politique de confidentialité
```

Installation locale

1. Cloner et installer les dépendances

```
bash  
  
git clone votre-repo  
cd yt-creator-tools  
pip install -r requirements.txt
```

2. Configuration

Copier `.env.example` vers `.env` et configurer :

```
bash
```

```
cp .env.example .env
```

Modifier `.env` :

```
FLASK_ENV=development
```

```
PORT=5000
```

```
YOUTUBE_API_KEY=votre_cle_api_youtube_ici
```

3. Lancer en local

```
bash
```

```
python app.py
```

L'application sera accessible sur `http://localhost:5000`

Déploiement sur Render

Étape 1 : Préparation du code

1. Assurez-vous que tous les fichiers sont dans votre repo Git
2. Poussez sur GitHub/GitLab :

```
bash
```

```
git add .
```

```
git commit -m "Initial commit"
```

```
git push origin main
```

Étape 2 : Créer un service sur Render

1. Allez sur render.com
2. Connectez-vous ou créez un compte
3. Cliquez sur "New +" > "Web Service"
4. Connectez votre repository GitHub/GitLab
5. Sélectionnez votre projet `yt-creator-tools`

Étape 3 : Configuration du service

Paramètres à configurer :

Champ	Valeur
Name	yt-creator-tools
Environment	Python 3
Build Command	pip install -r requirements.txt
Start Command	gunicorn app:app
Plan	Free (ou payant si besoin)

Étape 4 : Variables d'environnement

Dans la section **Environment Variables**, ajoutez :

Nom	Valeur	Notes
FLASK_ENV	production	Obligatoire
YOUTUBE_API_KEY	votre_clé_api	Optionnel, pour métadonnées complètes
PYTHONPATH	/opt/render/project/src	Recommandé

Étape 5 : Déploiement

1. Cliquez sur "Create Web Service"
2. Render va automatiquement :
 - Cloner votre repo
 - Installer les dépendances
 - Démarrer l'application
3. Attendez que le déploiement se termine (5-10 minutes)

Étape 6 : Vérification

Votre application sera accessible sur : `https://votre-app-name.onrender.com`

Testez les endpoints :

- `GET /` : Interface utilisateur
- `GET /api/test` : Test de l'API
- `POST /api/video-info` : Métadonnées vidéo
- `POST /api/subtitles` : Extraction de sous-titres

Configuration YouTube API (Optionnel)

Pour obtenir des métadonnées complètes :

1. Créer un projet Google Cloud

1. Allez sur [Google Cloud Console](#)
2. Créez un nouveau projet
3. Activez l'API YouTube Data API v3

2. Créer une clé API

1. Allez dans **APIs & Services** > **Credentials**
2. Cliquez sur "Create Credentials" > "API Key"
3. Copiez la clé générée

3. Restreindre la clé (recommandé)

1. Cliquez sur votre clé API
2. Sous **API restrictions** :
 - Sélectionnez "Restrict key"
 - Choisissez "YouTube Data API v3"
3. Sous **Application restrictions** :
 - Sélectionnez "HTTP referrers"
 - Ajoutez votre domaine Render

4. Ajouter la clé sur Render

1. Dans le dashboard Render de votre service
2. Allez dans **Environment**
3. Ajoutez `YOUTUBE_API_KEY` avec votre clé

Configuration Firebase (Optionnel)

Pour l'authentification complète :

1. Créer un projet Firebase

1. Allez sur [Firebase Console](#)
2. Créez un nouveau projet
3. Activez **Authentication**
4. Configurez les fournisseurs (Email, Google)

2. Configuration Web

1. Dans **Project Settings** > **General**
2. Ajoutez une application Web

3. Copiez la configuration JavaScript
4. Remplacez dans `static/index.html` :

```
javascript

const firebaseConfig = {
  apiKey: "votre_api_key",
  authDomain: "votre_projet.firebaseio.com",
  projectId: "votre_projet_id",
  // ... autres paramètres
};
```

3. Domaines autorisés

1. Dans **Authentication > Settings > Authorized domains**
2. Ajoutez votre domaine Render : `votre-app.onrender.com`

Mise à jour et maintenance

Redéploiement automatique

Render redéploie automatiquement à chaque push sur la branche `main`.

Logs et monitoring

1. Dans le dashboard Render
2. Onglet "**Logs**" pour voir les logs en temps réel
3. Onglet "**Metrics**" pour les performances

Mise à jour des dépendances

```
bash

pip list --outdated
pip install --upgrade package-name
pip freeze > requirements.txt
git commit -am "Update dependencies"
git push
```

Limites du plan gratuit Render

- **Inactivité** : L'app se met en veille après 15 minutes d'inactivité
- **Temps de build** : 500 heures/mois
- **Bande passante** : 100GB/mois
- **Stockage** : Éphémère, redémarrage = perte des fichiers

Dépannage

L'application ne démarre pas

Vérifiez les logs Render :

```
bash

# Erreurs communes :
# - Module non trouvé : vérifiez requirements.txt
# - Port incorrect : utilisez la variable PORT de Render
# - Timeout : augmentez les timeouts Gunicorn
```

Les sous-titres ne se téléchargent pas

1. Vérifiez que `youtube-transcript-api` est installé
2. Testez avec des vidéos ayant des sous-titres
3. Vérifiez les logs d'erreur

L'authentification Firebase ne fonctionne pas

1. Vérifiez la configuration Firebase dans le HTML
2. Ajoutez votre domaine Render aux domaines autorisés
3. Vérifiez la console des erreurs du navigateur

Support

Pour toute question ou problème :

- Ouvrir une issue GitHub
- Consulter les logs Render
- Vérifier la documentation de `youtube-transcript-api`

License

MIT License - voir le fichier LICENSE pour plus de détails.