

FileConverter CLI

<p align="center">

Dosyalarinizi yerel ortamda guvenli bir sekilde donusturun.

Belge, ses ve gorsel dosyalarini internet'e yuklemeden, tamamen yerel olarak farkli formatlara donusturun.

</p>

<p align="center">

</p>

Özellikler

• %100 Yerel • Dosyalariniz hicbir zaman internet'e yuklenmez

• Hızlı • Go ile yazilmis, optimize edilmiş performans

• Toplu Donusum • Worker pool ile paralel batch donusum

• Belge Formatlari • MD, HTML, PDF, DOCX, TXT

• Ses Formatlari • MP3, WAV, OGG, FLAC, AAC, M4A, WMA

• Gorsel Formatlari • PNG, JPEG, WEBP, BMP, GIF, TIFF

• Kullanici Dostu • Renkli cikti, progress bar, emoji ikonlar

• Shell Completion • Bash, Zsh, Fish, PowerShell destegi

Gereksinimler

• Go 1.21+ (derlemek için)

• FFmpeg (yalnizca ses donusumleri için gerekli)

FFmpeg Kurulumu (opsiyonel)

bash

macOS

brew install ffmpeg

Ubuntu/Debian

```
sudo apt install ffmpeg
```

Windows (Chocolatey)

```
choco install ffmpeg
```

ÖYš€ Kurulum

Go ile Kurulum

```
bash
```

```
go install github.com/melihgenel/fileconverter@latest
```

Kaynaktan Derleme

```
bash
```

```
git clone https://github.com/melihgenel/fileconverter.git
```

```
cd fileconverter
```

```
go build -o fileconverter .
```

ÖY“– Kullanım

Tekli Donusum

```
bash
```

Markdown â†’ PDF

```
fileconverter convert README.md --to pdf
```

Markdown â†’ HTML

```
fileconverter convert belge.md --to html
```

Markdown â†’ DOCX

```
fileconverter convert rapor.md --to docx
```

PDF â†’ Plain Text

```
fileconverter convert dosya.pdf --to txt
```

Gorsel donusum (kalite ayari ile)

```
fileconverter convert resim.png --to jpg --quality 90
```

Ses donusum

```
fileconverter convert muzik.mp3 --to wav
```

Cikti dizini belirtme

```
fileconverter convert dosya.md --to pdf --output ./cikti/
```

Cikti dosya adi belirtme

```
fileconverter convert dosya.md --to pdf --name sonuc
```

Toplu Donusum (Batch)

bash

Dizindeki tum MD dosyalarini PDF'e donustur

```
fileconverter batch ./belgeler --from md --to pdf
```

Alt dizinleri de dahil et

```
fileconverter batch ./belgeler --from md --to pdf --recursive
```

Ciktiyi farkli dizine yaz

```
fileconverter batch ./belgeler --from md --to html --output ./cikti/
```

Worker sayisini ayarla

```
fileconverter batch ./resimler --from png --to jpg --workers 8
```

Kalite ayari ile

```
fileconverter batch ./resimler --from png --to jpg --quality 85
```

On izleme (dry-run)

```
fileconverter batch ./belgeler --from md --to pdf --dry-run
```

Desteklenen Formatlari Goruntuleme

bash

Tum desteklenen formatlar

```
fileconverter formats
```

Belirli bir formattan yapilabilecek donusumler

```
fileconverter formats --from pdf
```

Belirli bir formata yapilabilecek donusumler

fileconverter formats --to docx

Ö“Š Desteklenen Donusumler

Ö“„ Belge Formatlari

Kaynak Hedef Formatlar
----- -----
MD HTML, PDF, TXT, DOCX
HTML TXT, MD
PDF TXT
DOCX TXT
TXT PDF, HTML, DOCX

ÖŽµ Ses Formatlari (FFmpeg gerektirir)

MP3, WAV, OGG, FLAC, AAC, M4A, WMA â€” tum formatlar arasi capraz donusum (42 yol)

Ö¼• Gorsel Formatlari

Kaynak Hedef Formatlar
----- -----
PNG JPG, BMP, GIF, TIFF
JPEG PNG, BMP, GIF, TIFF
WEBP PNG, JPG, BMP, GIF, TIFF
BMP PNG, JPG, GIF, TIFF
GIF PNG, JPG, BMP, TIFF
TIFF PNG, JPG, BMP, GIF

Toplam: 18 format, 78 donusum yolu

âŠ™• Global Secenekler

Flag Kisa Aciklama
----- ----- -----
--verbose -v Detayli cikti modu
--output -o Cikti dizini
--workers -w Paralel worker sayisi (varsayilan: CPU cekirdek sayisi)
--version Versiyon bilgisi
--help -h Yardim

ÖY•İ• Proje Yapisi

FileConverter/

```
â"œâ"€â"€ main.go          # Giriş noktası
â"œâ"€â"€ cmd/
â", â"œâ"€â"€ root.go       # Root komut, global flag'ler
â", â"œâ"€â"€ convert.go    # Tekli dönüşüm
â", â"œâ"€â"€ batch.go      # Toplu dönüşüm
â", â""â"€â"€ formats.go    # Format listesi
â"œâ"€â"€ internal/
â", â"œâ"€â"€ converter/
â", â", â"œâ"€â"€ converter.go # Interface + Registry
â", â", â"œâ"€â"€ document.go  # Belge dönüşümleri
â", â", â"œâ"€â"€ audio.go     # Ses dönüşümleri (FFmpeg)
â", â", â""â"€â"€ image.go     # Görsel dönüşümleri
â", â"œâ"€â"€ batch/
â", â", â""â"€â"€ pool.go      # Worker Pool
â", â""â"€â"€ ui/
â", â""â"€â"€ progress.go     # Progress bar, renkli çıktı
â"œâ"€â"€ go.mod
â""â"€â"€ go.sum
```

ÖY• Katkıda Bulunma

1. Fork yapın
2. Feature branch oluşturun (git checkout -b feature/yeni-format)
3. Değişikliklerinizi commit edin (git commit -m 'Yeni format desteği eklendi')
4. Branch'e push edin (git push origin feature/yeni-format)
5. Pull Request acın

Yeni Converter Ekleme

internal/converter/ dizininde yeni bir dosya oluşturun ve Converter interface'ini implemente edin:

```
go
```

```
package converter
```

```
type MyConverter struct{}
```

```
func init() {  
    Register(&MyConverter{})  
}
```

```
func (c *MyConverter) Name() string { return "My Converter" }  
func (c *MyConverter) SupportsConversion(from, to string) bool { /* ... */ }  
func (c *MyConverter) SupportedConversions() []ConversionPair { /* ... */ }  
func (c *MyConverter) Convert(input, output string, opts Options) error { /* ... */ }
```

ÖY“, Lisans

MIT License â€” detaylar için LICENSE dosyasına bakın.