

# Electron ile Masaüstü Uygulama Geliştirme – Temel Özellikler

---

Eren Altay

20360859023

3. Sınıf

09.05.2024



# İçindekiler

1. Giriş
  - Nedir
  - Avantajları
  - Dezavantajları
  - Kimler Kullanıyor
  - Alternatifler
2. Genel Uygulama Yapısı
  - Main ve Renderer Süreçler
  - Preload scriptler
  - IPC
3. Electron Uygulamalarını Paketleme
  - Electron Forge kullanımı

# Giriş

Nedir, Avantajlar ve Dezavantajlar,  
Kimler Kullanıyor, Alternatifler

# Nedir?

Electron, web teknolojilerini kullanarak Windows, macOS ve Linux için masaüstü uygulamaları geliştirmenizi sağlayan bir frameworktür. JavaScript, HTML ve CSS gibi web teknolojilerini kullanır ve Chromium ile Node.js gibi araçları temel alır.

# Avantajları

**WEB Teknolojileri**  
HTML, CSS ve JS başta olmak üzere web teknolojilerini kullanır.

**Cross-Platform**  
Windows, macOS ve Linux gibi farklı işletim sistemlerinde çalışabilen uygulamalar geliştirmenizi sağlar.

**Açık Kaynak**  
Electron açık kaynaklıdır ve geniş bir kitlesi vardır.

# Dezavantajları

## **Kaynak Kullanımı**

Electron'un kaynak kullanımı diğer masaüstü uygulama geliştirme platformlarından daha yüksektir.

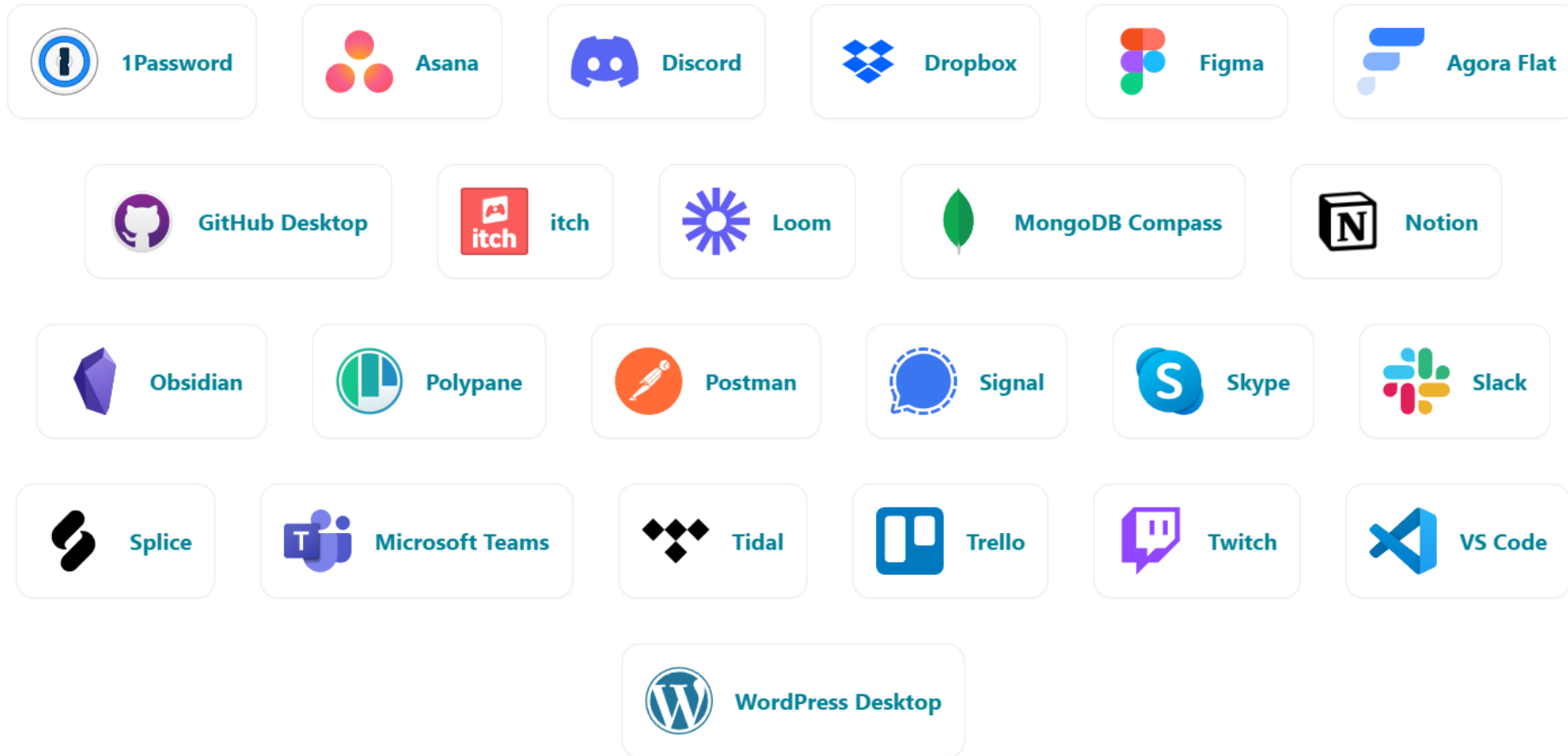
## **Uygulama Boyutu**

Electron ile geliştirilen uygulamaların boyutu fazladır.

## **Bağımlılıklar**

Electron projeleri bakımı ve dosya boyutunu artıran bağımlılıklara sahip olabilir.

# Kimler Kullanıyor?



# Alternatifler

---

Tauri: HTML, CSS, JS ve RUST programlama dillerine dayanır. Kaynak kullanımı Electron'dan daha azdır.

---

Native PHP: HTML, CSS, JS ve PHP kullanır.

---

Flutter: Flutter ile masaüstü uygulamalar geliştirilebilir.

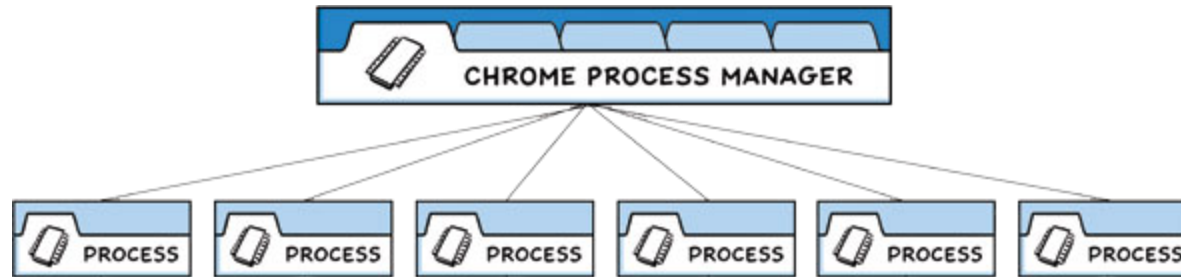


# Genel Uygulama Yapısı

Process Modeli, Main Process,  
Renderer Process, Preload Scriptler,  
ipc, ipcMain ve ipcRenderer

# Electron Process Modeli

- Electron, Chromiumun multi-process mimarisini kullanılır.



- Electron uygulamalarında 2 ana süreç bulunur: Main ve Renderer Processler

# Main Process

- Electron uygulamasının yaşam döngüsünü kontrol eder. Bir Electron uygulamasında yalnızca bir ana süreç olabilir.
- Electron API'lerine tam erişime sahiptir ve doğrudan işletim sistemiyle etkileşim kurabilir.
- Ana süreç uygulamanın tüm pencerelerini oluşturabilir ve yönetebilir.

# Renderer Process

- Bu süreçler web sayfalarını oluşturur. Her electron penceresi (BrowserWindow) kendi renderer sürecine sahiptir. DOM, Node.js ve Electron API'lerine erişebilirler.
- İşletim sistemiyle doğrudan etkileşim kuramazlar.

# Preload Scriptler

- Renderer processlerden önce çalıştırılırlar.
- Node.js ve Electron API'lerine renderer processlerden daha fazla erişimi vardır.
- Güvenlik kontrolleri, API erişimleri, veri ön yükleme gibi işlemler preload scriptlerde yapılabilir.

# Preload Scriptlerin Erişebildiği API'ler

- Electron 20'den itibaren preload scriptlerin tüm Node.js özelliklerine erişimi yoktur.
- Erişimi olan API'ler:
  - Electron Modülleri: Renderer ve Process modülleri
  - Node.js Modülleri: events, timers, url
  - Buffer, process, clearImmediate, setImmediate

# IPC(Inter-Process-Communication)

Electron'da IPC (Inter-Process Communication), Main ve Renderer süreçleri arasında iletişim kurmayı sağlar. Electron, bu iletişimi sağlamak için iki modül sunar: ipcMain ve ipcRenderer.

# ipcMain Modülü

ipcMain Modülü: Ana süreçte kullanılır. Renderer süreçlerinden gelen asenkron ve senkron mesajları dinlemek için kullanılır. Asenkron mesajlar için ipcMain.on metodu, senkron mesajlar için ipcMain.handle metodu kullanılır.



# ipcRendererer Modülü

Renderer süreçlerinde kullanılır. Ana sürece asenkron ve senkron mesajlar göndermek için kullanılır. Asenkron mesajlar için `ipcRenderer.send` metodu, senkron mesajlar için `ipcRenderer.invoke` metodu kullanılır.

# Electron Uygulamalarını Paketleme

Paketleme ve Dağıtım, Electron Forge  
Kullanımı

# Paketleme ve Dağıtım

- Electron paketleme ve dağıtım için çekirdek modüllerinde bir araç sunmaz.
- Paketleme ve dağıtım için ekstra paketler kullanılmalıdır.
- Electron Forge bu araçlardan birisidir.
- Electron Packager ve Electron Builder de kullanılabilir.

# Electron Forge Kullanımı

```
npm install --save-dev @electron-forge/cli  
npx electron-forge import
```

package.json

```
//...  
"scripts": {  
  "start": "electron-forge start",  
  "package": "electron-forge package",  
  "make": "electron-forge make"  
},  
//...
```

```
npm run make
```

# Kaynaklar

- <https://www.electronjs.org/docs/latest/>
- <https://www.electronforge.io>
- <https://chiragagrawal65.medium.com/how-to-import-ipc-renderer-in-renderer-process-component-26fef55fa4b7>
- [https://en.wikipedia.org/wiki/Electron\\_\(software\\_framework\)](https://en.wikipedia.org/wiki/Electron_(software_framework))

# Sorularınız?

# Teşekkürler