

GLIEDERUNG

1. Electron
2. ECMAScript6
3. NodeJs

ELECTRON

- Open Source Library von GitHub
- Desktop-Applikation erstellen mit HTML, CSS, JavaScript
- Applikationen laufen unter Mac, Windows und Linux

ELECTRON IPC

ZWEI ARTEN VON PROZESSEN:

- Ein Main-Prozess
- n Renderer-Prozesse

ELECTRON IPC

IPC RENDERER MODULE:

Kommunikation vom Renderer- zum Main Prozess

IPC MAIN MODULE:

Kommunikation vom Main- zum Renderer-Prozess

```
ipcRenderer.send('asynchronous-message', 'ping')

ipcMain.on('asynchronous-message', (event, arg) => {
  console.log(arg) // prints "ping"
  event.sender.sendSync('synchronous-message', 'pong')
})

ipcRenderer.on('synchronous-message', ...
```

APP MODULE

- Läuft im Main-Prozess
- Verwaltet den Lifecycle der Applikation
- Events können mehrere Parameter entgegennehmen
- Jedes Events gibt unterschiedliche Parameter zurück

```
const {app} = require('electron')
app.on('event', (... parameters ...) => {
  // do something
})
```

APP MODULE EVENTS

- ready
- window-all-closed
- before-quit
- open-file
- browser-window-created
- ...

BROWSERWINDOW

- Läuft im Main-Prozess
- Verwaltet Fensterabhängige Events
- Parent-Child Konzept
- Options erlauben Fensterkonfiguration

ECMAScript

- standardisierter Sprachkern von JavaScript
- ECMAScript 2015 (ECMAScript 6):
 - Optionale neue klassenbasierte Objektorientierung als Syntactic Sugar
- Module
- Und vieles weiteres

TYPESCRIPT

- entwickelt von Microsoft
- Superset über JavaScript, welches zu ES3 (optional ES5) transpiliert
- Statische Typisierung
- Interfaces

NODEJS

- JavaScript-Laufzeitumgebung für Server-Anwendungen
- Open Source, cross-plattform
- veröffentlicht 2009 von Ryan Dahl
- benutzt JavaScript-Engine V8 von Google Chrome
- Anwendungen auch in CoffeeScript, Dart oder TypeScript möglich
- wird u.a. eingesetzt von: IBM, LinkedIn, Microsoft, Netflix, PayPal, SAP, Yahoo, Cisco, ...
- Paketmanager zum Installieren und Verwalten von Modulen
- schreiben eigener Module möglich

NODEJS

- arbeitet standardmäßig mit nur einem Thread
- mit libuv kann zusätzlich Thread-Pool vorgehalten werden
- event-driven-Architektur erlaubt auch ohne Threading hochskalierbare Web Server
- asynchroner I/O
- Funktionen laufen nicht-blockierend (parallel) ab; Callbacks signalisieren, wenn sie fertig oder fehlgeschlagen sind