

Desktop Apps med HTML, CSS og JS

**En introduksjon til Electron
BartJS 10.11.2015**

275 dager på tur

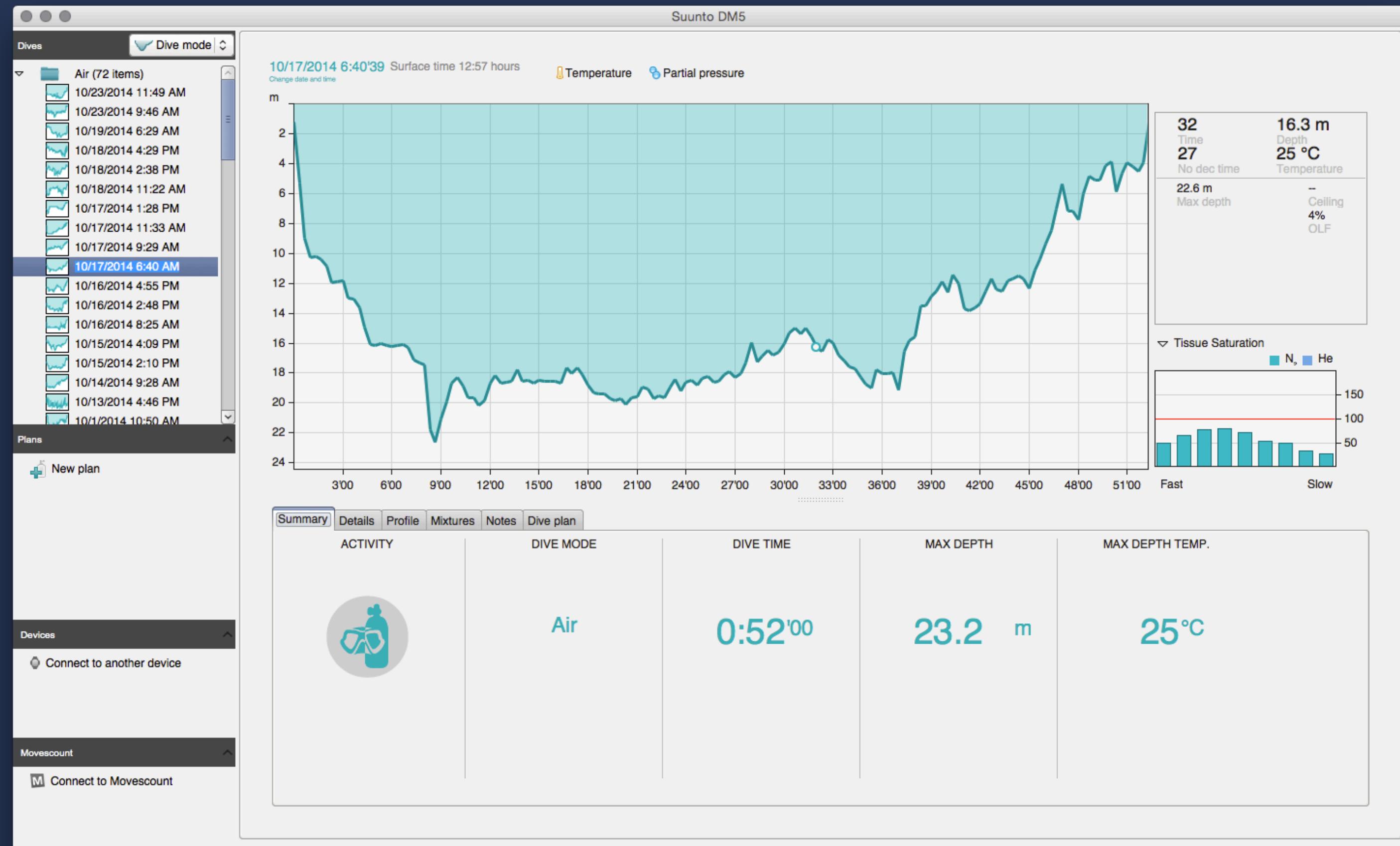




Rob Allen

Suunto D6









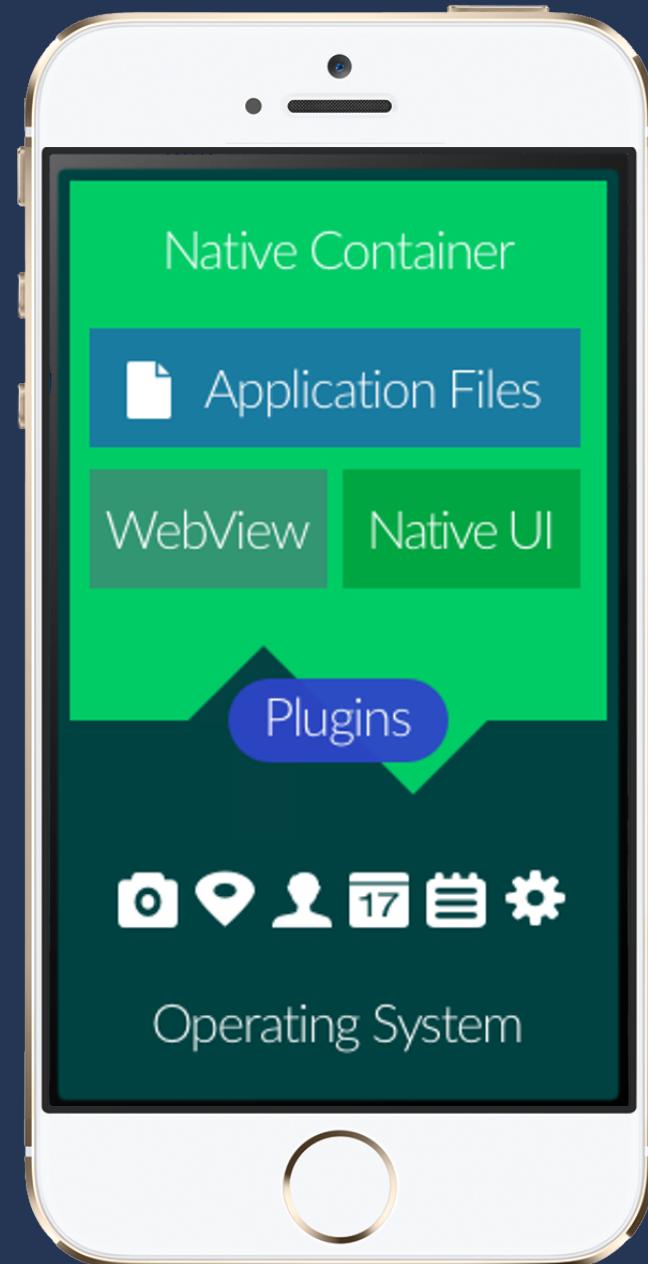
.config					
	Name	Date Modified	Size	Kind	
Favorites	► configstore	17 Aug 2015 21:24	--	Folder	
	► git	Today 18:32	--	Folder	
	► gtk-2.0	Today 08:22	--	Folder	
	► inkscape	25 Oct 2013 08:38	--	Folder	
	► NuGet	7 Nov 2013 10:53	--	Folder	
	► stetic	19 May 2013 13:11	--	Folder	
	▼ Suunto	7 Aug 2015 09:46	--	Folder	
	▼ Suunto DM5	12 Jun 2015 24:16	--	Folder	
	▼ 1.2.42.5696	Today 08:24	--	Folder	
	► DM4.db	Today 08:24	362 KB	Document	
	► DM5 OSX_1x2x42_2x2x15.log	Today 08:22	Zero bytes	Log File	
	► fluentconfig	2 Jun 2015 19:56	207 KB	TextEd...ument	
	► restoredm4.backup	14 Aug 2015 12:57	46 KB	Document	
	► SuuntoDM5_6u25dg41.7io.log	Today 08:22	164 bytes	Log File	
	► SuuntoDM5_p94b36hd.uge.log	Today 08:22	164 bytes	Log File	
	► xbuild	19 May 2013 13:11	--	Folder	
Devices					
	► mac-jonasfolleso				
	► Macintosh HD				

Fiks det sjø!

**WEB,
HYBRID,
NATIVE?**

Hybrid mobil apps

Instead of targeting a mobile browser, hybrid applications target a WebView hosted inside a native container.



Hybrid Desktop Apps

divmanager-edit

untitled package.json - divem gulpfile.js package.json - nwjsde index.html Desktop Apps med H package.json - src/nw

```
> .git  
> cache  
> icons  
> node_modules  
> output  
> src  
  > electronapp  
  > nwjsapp  
  > shared  
    .DS_Store  
    .gitignore  
> templates  
> webapp  
  .DS_Store  
  .gitignore  
  .npmignore  
  gulpfile.js  
  package.json  
  README.md
```

> 2015-09-04 - BEKK Fagdag

```
> Images  
  .DS_Store  
  Atom Screenshot.png  
  D6.jpg  
  DM5-Dykk.png  
  DM5-Fridykk.png  
  DM5.png  
  Dykking.mp4  
  electron-icon.png  
  electron-logo.svg  
  Fridykkning.jpg
```

```
59  ! [fit] (Images/Slack.png)  
60  
61  ---  
62  
63  # NW.js  
64  
65  *Nw.js lets you call all Node.js modules directly from DOM and enables a new way of  
   writing applications with all Web technologies.*  
66  
67  ! [45%] (Images/nw-icon.png)  
68  
69  
70  ---  
71  
72  # package.json  
73  
74  ```json  
75  {  
76    "name": "SuuntoDMEEditor",  
77    "main": "index.html",  
78    "window": {  
79      "title": "Suunto Dive Manager Editor",  
80      "toolbar": false,  
81      "width": 1024,  
82      "height": 720  
83    }  
84  }
```



Slack

BEKK

jonas.folleso

CHANNELS

fsharp

general

random

trondheim

trondheim-dotnet

trondheim-frontend

trondheim-hw

+46 More...

DIRECT MESSAGES

slackbot

Alexander Hanssen

Eivind Sorteberg

Emil Mork

Jørn Vagnby Lillesand

Ken Grønnbeck

Mikael Brevik

Odd Morten Sveås

Ole Christian Langfjæræn

torgeir

Øystein Samuelsen

+265 More...

PRIVATE GROUPS

da

dualog

eir

New private group...

#general

276

Search

August 28th

trenger noe som ikke er Google Analytics. Per na bruker kunden Adobe Analytics, og det er seriøst jævlig a jobbe mea.

kjersti 9:57 AM
joined #general

Safurudin Mahic 10:00 AM
@bendik for vår bruk er det yay. Men hva er bruksscenarioet ditt?

bendik 10:02 AM
Ideelt sett hadde de brukt Google Analytics, men de kan ikke på grunn av måten Google bruker dataene på. De har ganske strenge krav til dette, hvis jeg har forstått dem riktig. Så de trenger noe hvor det enten garanteres at ingen andre enn kunden kan se dataene (ikke anonymisert eller noe), eller noe som installeres på kundens systemer.

August 31st

Jørn Vagnby Lillesand 8:05 AM
Mine venner, hvem er det som er kassererer i Spirit igjen nå?

Øyvind Midtbø 8:06 AM
Økonomiansvarlig er @lyckander

Jørn Vagnby Lillesand 8:07 AM
Thanx 😊

Ole Christian Langfjæræn 12:25 PM
Det jeg synes er litt kjedelig på Bekk-slacken er at historikken til kanalene forsvinner så fort at "it makes my head spin". Trumpstyle.
Regner med det er fordi internsystemene aggressivt pusher meldinger til diverse kanaler? Kunne det i så fall vært interessant å bruke en egen Slack til de systemene?
 2

Eivind Bergstøl 12:25 PM
+1
Det er 10000 meldinger vi har gratis elns?

Magnus Eiell 12:26 PM

The screenshot displays a user interface for a graphic design application, likely Sketch, showing a UI component for a food-related website.

General Panel:

- Name: Homepage
- Synced: 09/09/2015
- Project: My Blog
- Resolution: 2048x1023
- Format: Sketch
- Color: RGB

Sharing Options:

- Invite to Team
- Copy Public Link

Fonts Panel:

- Myriad Pro
 - Regular
 - Italic
- Futura LT
 - Bold
 - Regular

UI Component Preview:

A UI component is displayed over a background image of green beans. The component includes:

- Text:** "FIND DELICIOUS FOOD" (Large, bold, white font)
- Text:** "or simply browse the world's most inspiring accomodations!" (Medium, white font)
- Text:** "JOIN NOW" (Large, bold, white font, inside a red button)
- Dimensions:** X: 293 Y: 102 W: 272px H: 164px

Properties Panel (Top Right):

- Icon: T
- Name: Find Delicious Fo...
- Size: 272px x 164px
- Copy text

Code Panel (Bottom Right):

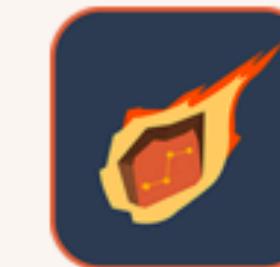
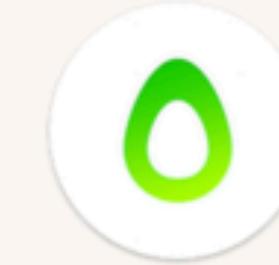
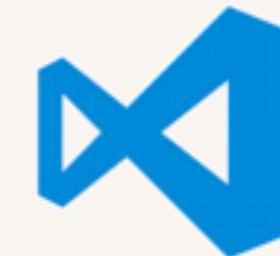
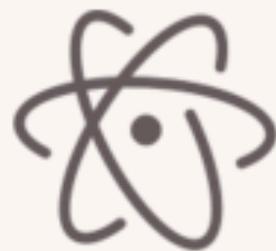
```
width: 272px;  
height: 164px;  
color: #fff;  
font-family: FuturaLT;  
font-size: 72px;  
font-weight: 700;  
line-height: 30px;
```

Export Panel (Bottom Right):

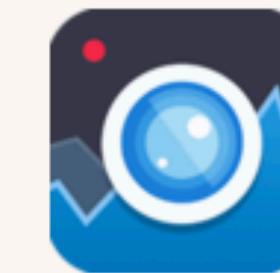
- Define Export Rules
- Quick Export

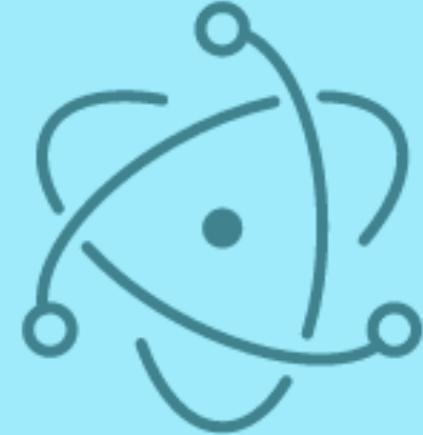
Zoom and View Controls:

- 100% +
- View mode icons



jibo





ELECTRON

Build cross platform desktop apps with web technologies

Formerly known as Atom Shell. Made with ❤ by GitHub.

Electron

Use HTML, CSS, and JavaScript with Chromium and Node.js to build your app.





DÉMO

Installasjon

```
npm install -g electron-prebuilt
```

package.json

```
{  
  "name"      : "hello-world",  
  "version"   : "0.0.1",  
  "main"      : "main.js"  
}
```

main.js

```
var app = require('app');
var BrowserWindow = require('browser-window');

var mainWindow = null;

app.on('ready', function() {
  mainWindow = new BrowserWindow({width: 800, height: 600});
  mainWindow.loadUrl('file://' + __dirname + '/index.html');

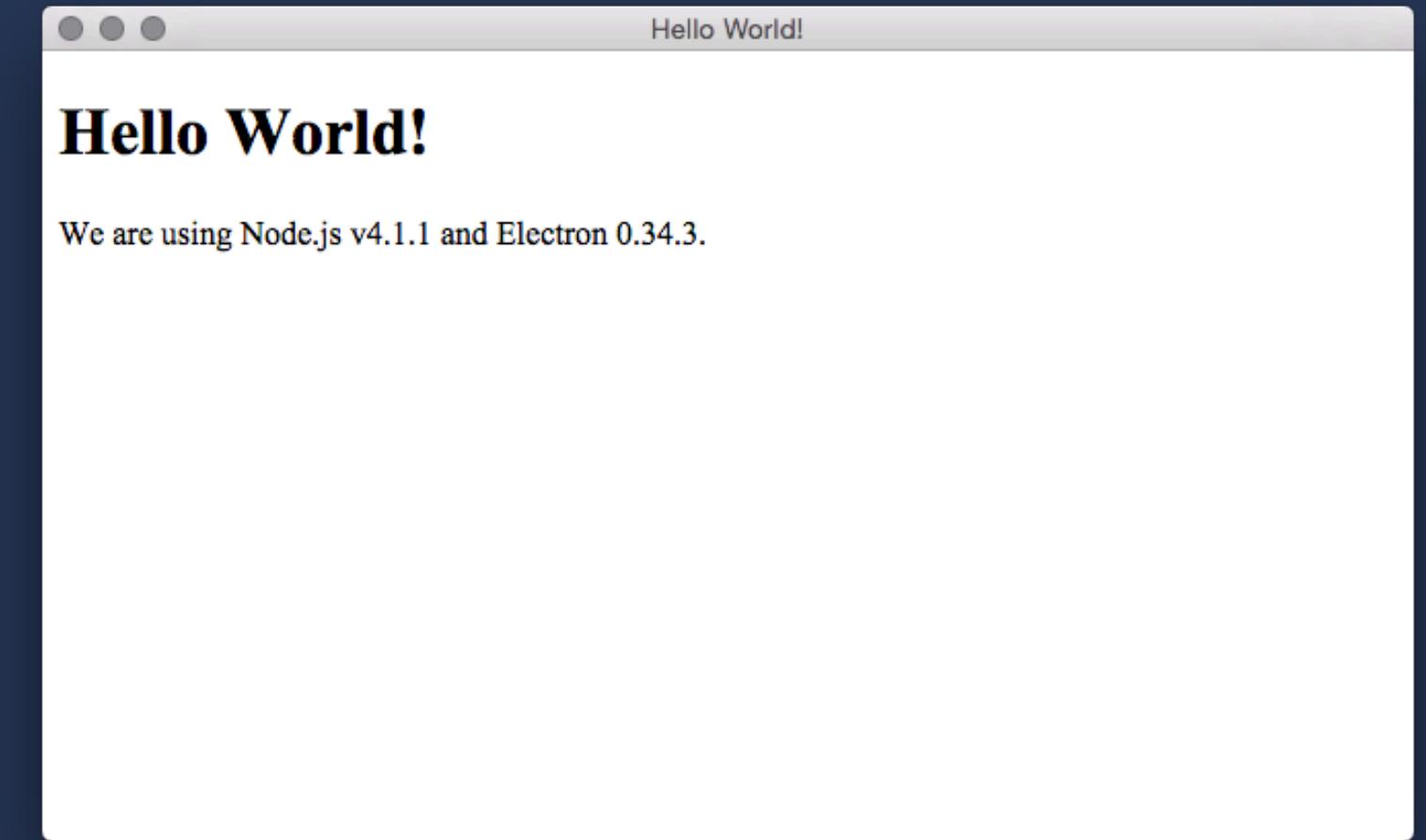
  mainWindow.on('closed', function() {
    mainWindow = null;
  });
});
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    We are using Node.js <script>document.write(process.version)</script>
    and Electron <script>document.write(process.versions['electron'])</script> .
  </body>
</html>
```

Kjøring

```
$ electron .
```



	13.10.2014	00:36:20	7	M
	01.10.2014	00:45:00	22	M
	01.10.2014	00:49:00	26	M
	28.09.2014	00:47:40	20	M
	28.09.2014	00:50:40	20	M
	27.09.2014	00:33:20	18	M
	27.09.2014	00:44:20	23	M
	26.09.2014	00:31:40	12	M
	24.09.2014	00:00:54	12	M
	24.09.2014	00:03:19	12	M
	24.09.2014	00:00:47	9	M
	24.09.2014	00:01:18	24	M
	24.09.2014	00:01:26	27	M
	24.09.2014	00:01:09	16	M

SELECTED DIVE

DIVE MODE



TIME IN

DEMO
MAX DEPTH 20 m
TEMP 23 °C

MAX DEPTH

20 m

DURATION

00:53:40 h

DIVE IN SERIES

2

COMPUTER

D9

SAVE CHANGES

Native Moduler

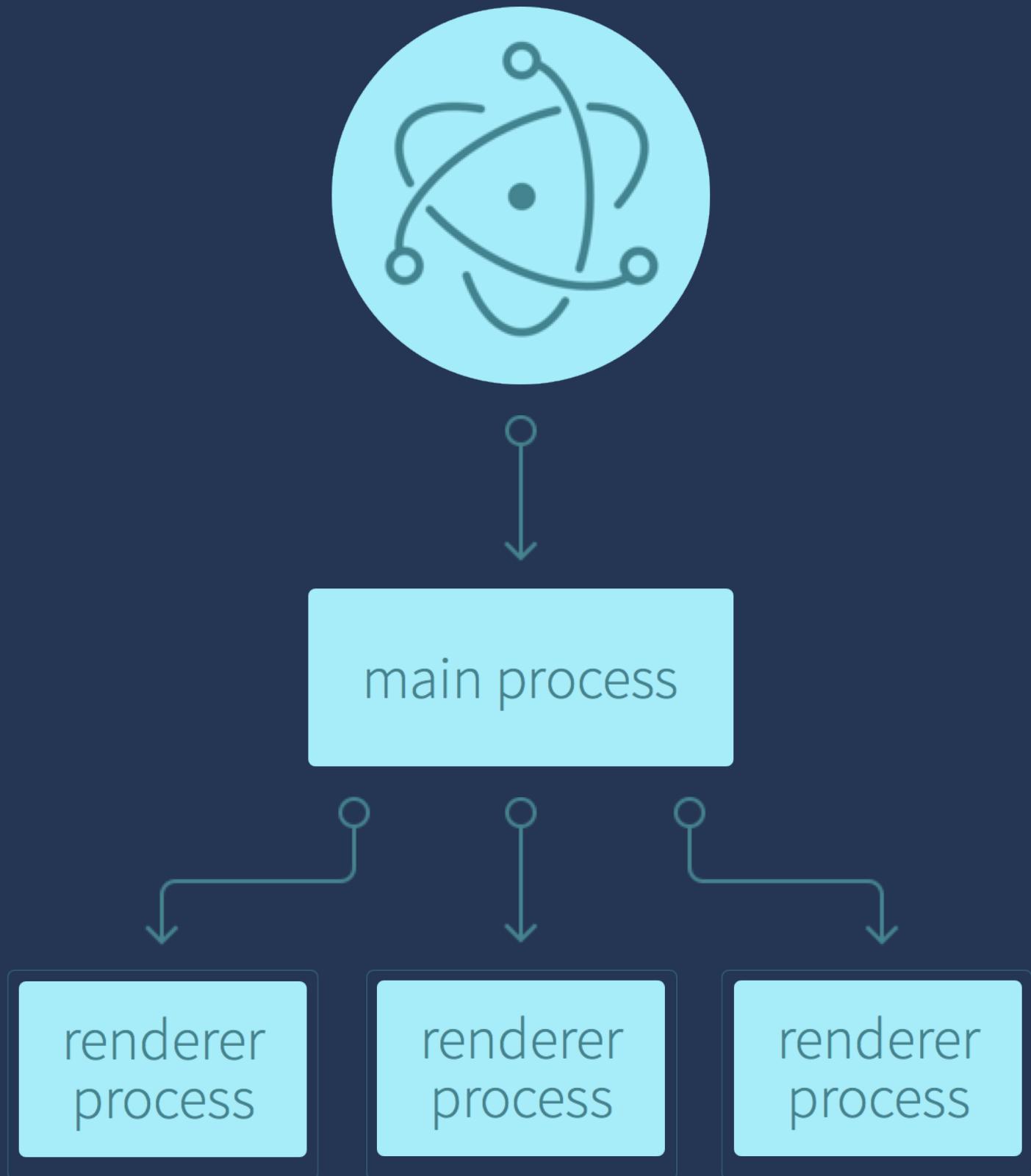
Oppdatering av dykk

```
var sqlite3 = require('sqlite3').verbose();

var saveDive = function (dbPath, dive) {
  var db = new sqlite3.Database(dbPath);
  db.serialize(function () {
    var stmt = db.prepare('UPDATE Dive SET Mode = (?) WHERE DiveId = (?)');
    stmt.run(dive.Mode, dive.DiveId);
    stmt.finalize();
  });
  db.close();
}

module.exports.saveDive = saveDive;
```

To prosesstyper



- Main process
 - Main script fra `package.json`
- Render process
 - En for hvert `BrowserWindow`
 - Ikke sandbox som vanlig nettleser, men full tilgang til Node.js

Kommunikasjon mellom prosesser

- *ipc* (inter-process communication)
- *remote* (remote objects)

remote

Lar deg enkelt opprette objekter fra renderer-prosess som lever på main-prosess

```
var remote = require('remote');
var BrowserWindow = remote.require('browser-window');
```

```
var win = new BrowserWindow({ width: 800, height: 600 });
win.loadUrl('https://github.com');
```



Meldinger mellom renderer-prosess og main-prosess

```
// In main process.  
var ipc = require('ipc');  
ipc.on('asynchronous-message', function(event, arg) {  
  console.log(arg); // prints "ping"  
  event.sender.send('asynchronous-reply', 'pong');  
});
```

```
// In renderer process (web page).  
var ipc = require('ipc');  
ipc.on('asynchronous-reply', function(arg) {  
  console.log(arg); // prints "pong"  
});  
ipc.send('asynchronous-message', 'ping');
```

MENUS!

Menu & MenuItem

Menyer kan lages gjennom *Menu* og *MenuItem* klassene...

```
var menu = new Menu();
menu.append(new MenuItem({
  label: 'MenuItem1',
  click: function() {}
}));
menu.append(new MenuItem({ type: 'separator' }));
menu.append(new MenuItem({
  label: 'MenuItem2',
  type: 'checkbox',
  checked: true
}));
```

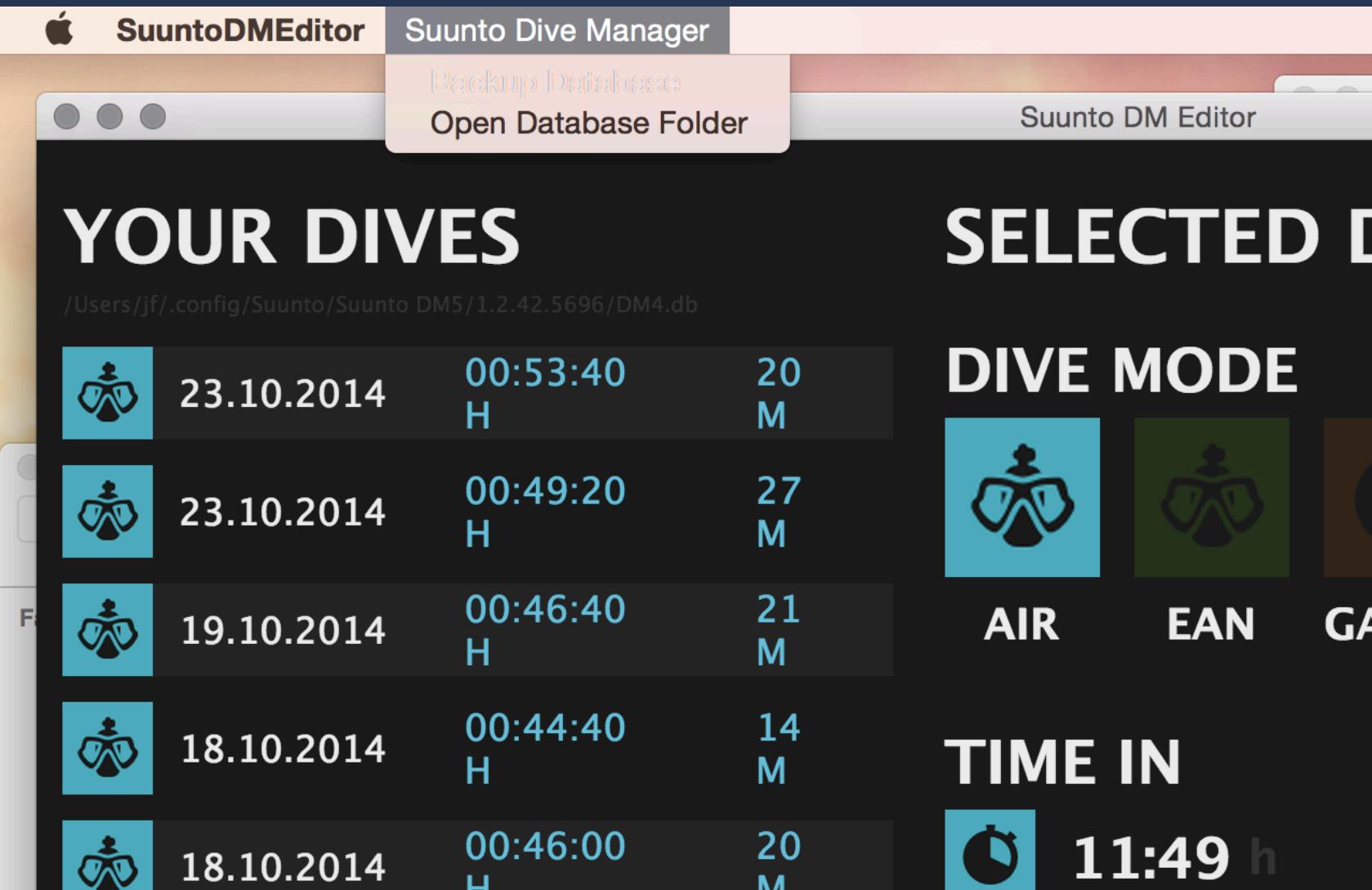
Menu & MenuItem

... eller via templates

```
var template = [ {  
    label: "MenuItem1", click: function() {} } ,  
    { type: "separator" } ,  
    { label: "MenuItem2", type: "checkbox", checked: true }  
] ;
```

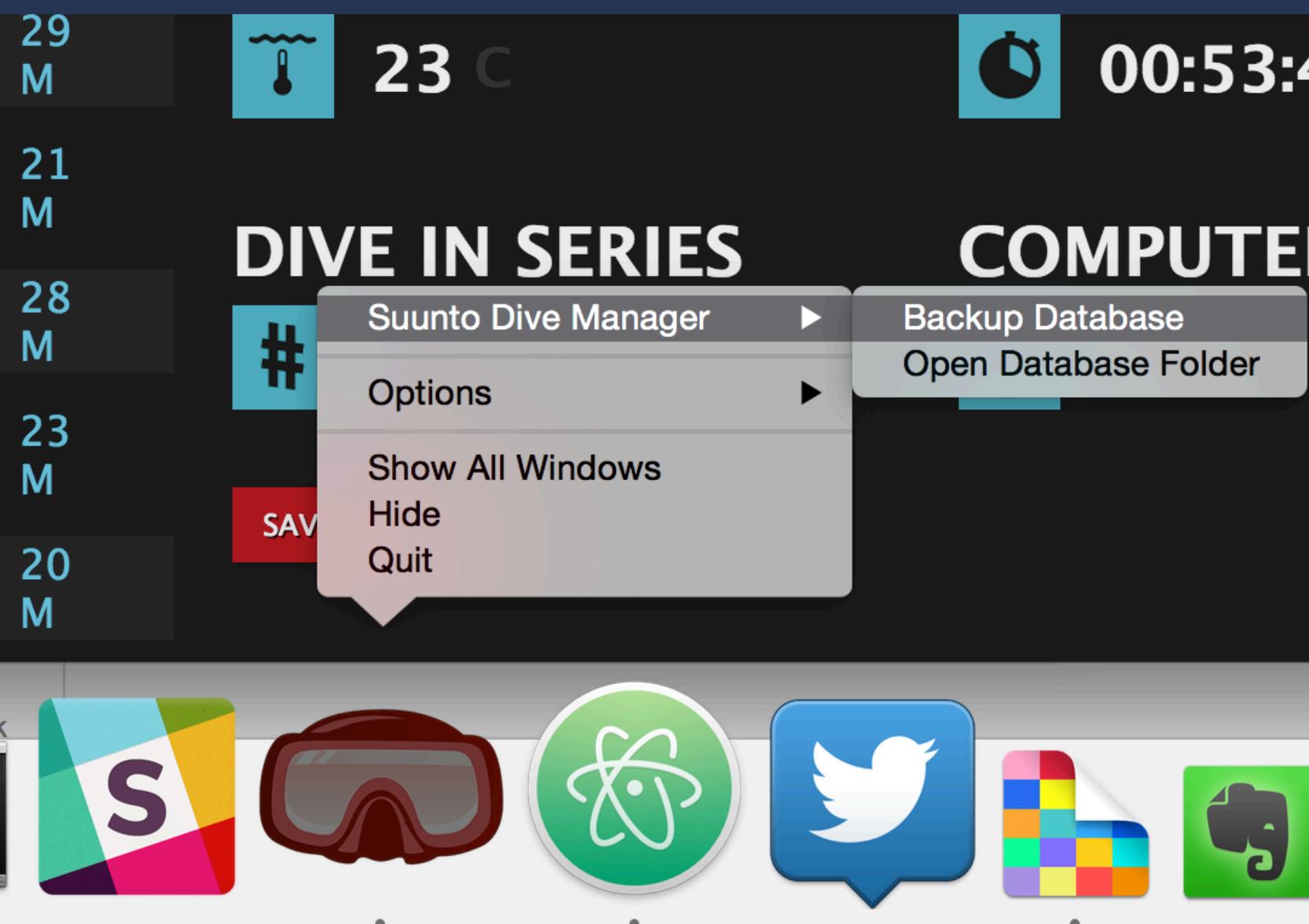
```
var menu = Menu.buildFromTemplate(template);
```

Application Menu

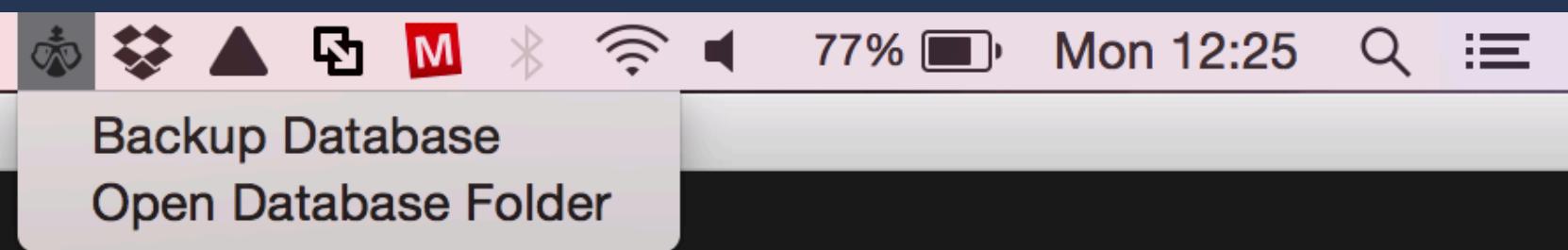


```
Menu.setApplicationMenu(menu);
```

Dock Menu



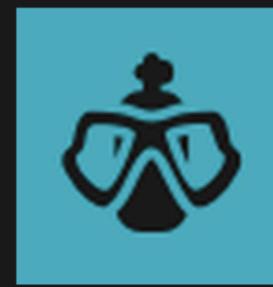
```
app.dock.setMenu(menu);
```



Tray Menu

SELECTED DIVE

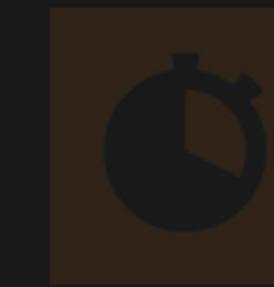
DIVE MODE



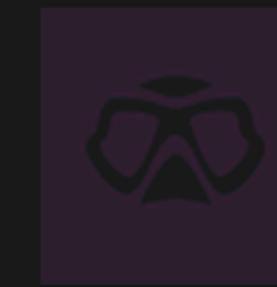
AIR



EAN

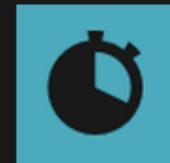


GAUGE



FREE

TIME IN



11:49 h

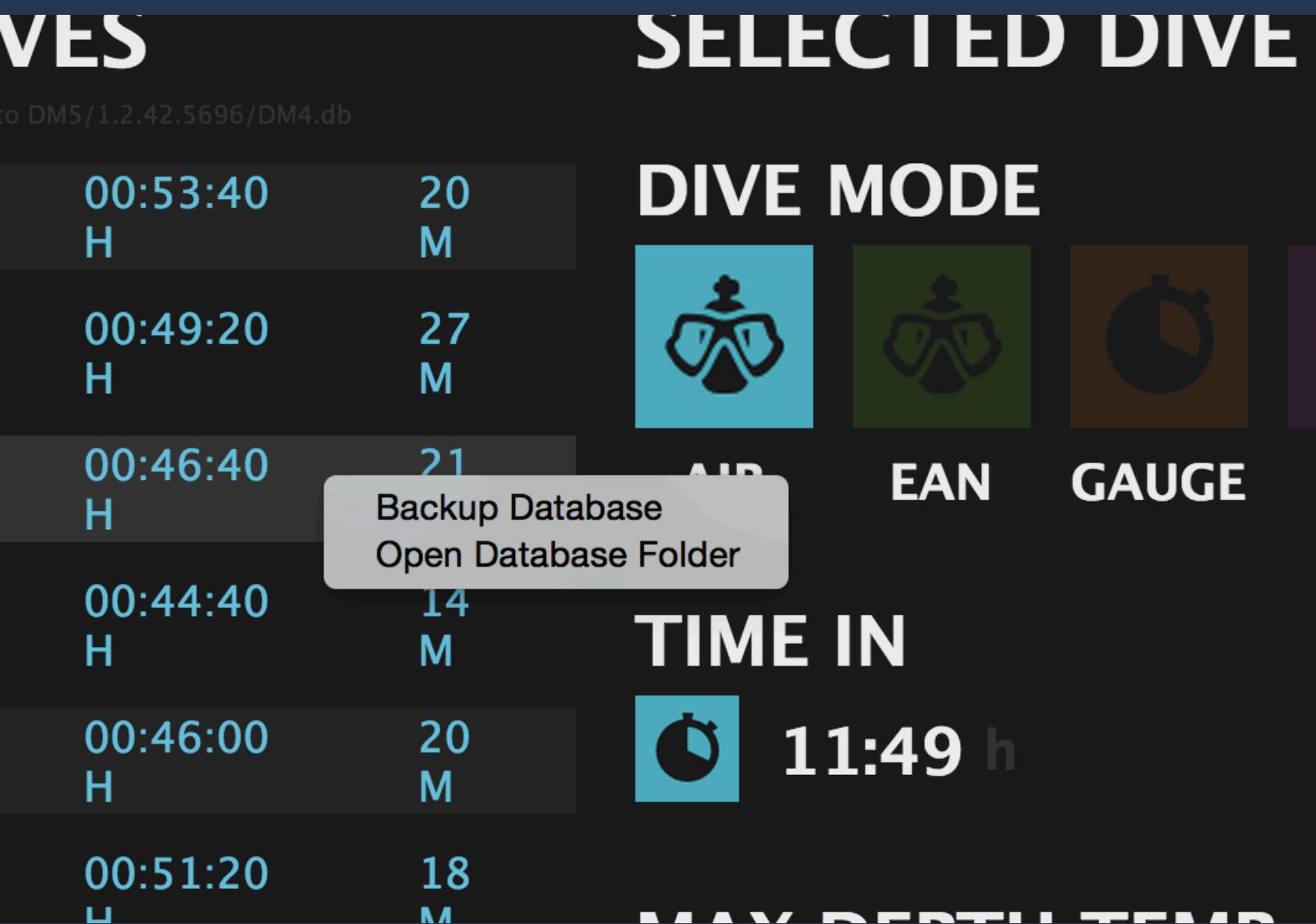
MAX DEP



20 m

```
appIcon = new Tray('images/trayicon.png');
appIcon.setToolTip('SuuntoDMEditor');
appIcon.setContextMenu(menu);
```

Context Menu



```
var Menu = remote.require('menu');

// ... template definition

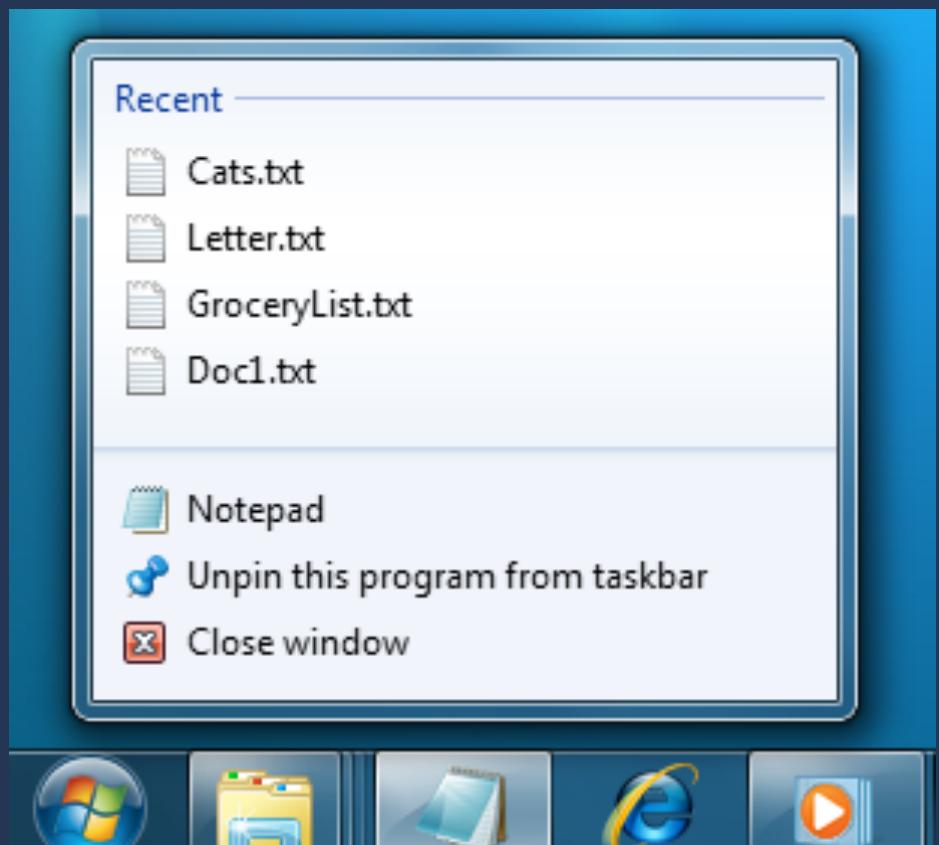
var menu = Menu.buildFromTemplate(template);

window.addEventListener('contextmenu', (e) => {
  e.preventDefault();
  menu.popup(remote.getCurrentWindow());
}, false);
```

DÉMO

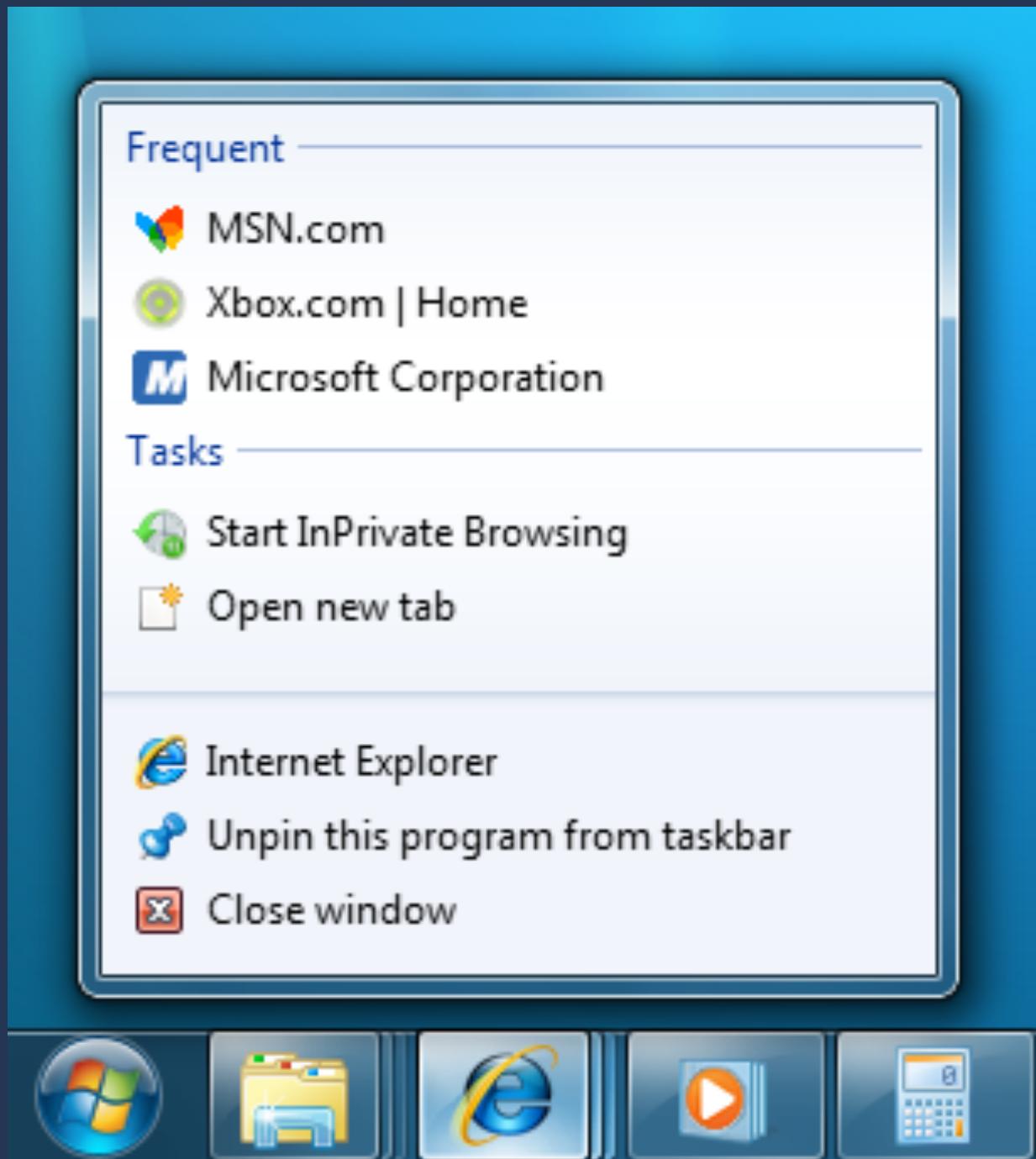
Windows Integrasjon

Recent Documents Jump List



```
app.addRecentDocument(' /Users/USERNAME/Desktop/work.type');
```

Windows Integrasjon



User Tasks Jump List

```
app.setUserTasks( [  
    {  
        program: process.execPath,  
        arguments: '--new-window',  
        iconPath: process.execPath,  
        iconIndex: 0,  
        title: 'New Window',  
        description: 'Create a new window'  
    }  
]);
```

DISTRIBUSJON

Sammen med Electron

Plasser appen din i en mappe kalt *app* under Electron sin resource-mappe

- OS X: *Electron.app/Contents/Resources*
- Linux og Windows: *resources/*

På OS X

electron/Electron.app/Contents/Resources/app/

- └── package.json
- └── main.js
- └── index.html

På Windows og Linux

electron/resources/app

- └── package.json
- └── main.js
- └── index.html

Kjøring av distribusjon

Kjør *Electron.app* (eller *electron* på Linux og *Electron.exe* på Windows) og Electron vil starte appen din.

Electron-mappen blir distribusjonen til sluttbruker.

Rebranding Electron Binary

- Linux: Endre navn på *electron* til hva du vil
- Windows: Endre navn på *Electron.exe* til hva du vil, og oppdater ikoner og metadata ved hjelp av verktøy som ResEdit.
- Mac: Endre navn på *Electron.app* og oppdatering av *.plist*-filer
- Alternativt: bygge fra source eller verktøy som *electron-packager*

electron-packager

Verktøy som pakker appen din og gjør rebranding av Electron

electron-packager

```
  .
  SuuntoDMEditor
  --platform=darwin
  --arch=all
  --version=0.34.3
  --out=../../output
  --icon=icons/main.icns
```

electron-packager

Verktøy som pakker appen din og gjør rebranding av Electron

```
"scripts": {  
  "build-mac": "electron-packager . SuuntoDMEditor --platform=darwin --arch=all --version=0.34.3 --out=../../output --icon=icons/main.icns",  
  "build-win": "electron-packager . SuuntoDMEditor --platform=win --arch=all --version=0.34.3 --out=../../output --icon=icons/main.ico"  
}
```

**AUTOMATISK
OPPDATERING**



Squirrel

Squirrel is an OS X framework focused on making application updates as safe and transparent as updates to a website.

- Mac & Windows
- autoUpdater er wrapper rundt Squirrel.Mac
- Mac Apps må være signert



autoUpdater

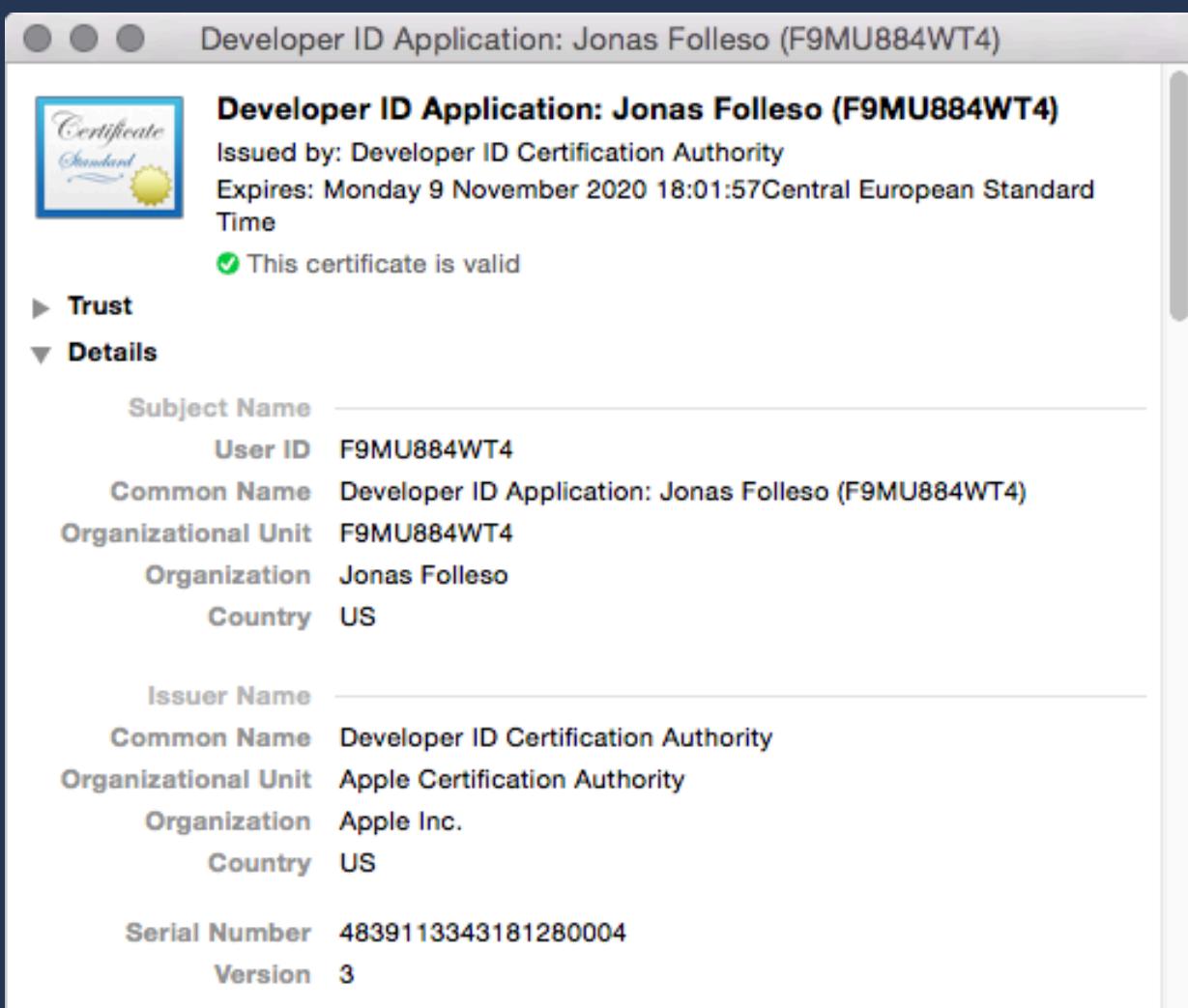
```
var autoUpdater = require('auto-updater');

autoUpdater.on('checking-for-update', () => { ... });
autoUpdater.on('update-available', () => { ... });
autoUpdater.on('update-not-available', () => { ... });
autoUpdater.on('update-downloaded', () => { ... });
autoUpdater.on('checking-for-update', () => { ... });

autoUpdater.setFeedUrl(feedUrl);
autoUpdater.checkForUpdates();
autoUpdater.quitAndInstall();
```

Signering av app

Må ha Apple Developer Program Membership (kr 779,-)



Signering av app

electron-packager kan signere under pakking

```
electron-packager . . . --sign=\"Developer ID Application: Jonas Follesø (F9MU884WT4)\"
```

Nuts

Nuts is a simple (and smart) application to serve desktop-application releases.

It uses GitHub as a backend to store assets, and it can easily be deployed to Heroku as a stateless service.

Mac App Store

- Electron v0.34.0 introduserte mulighet for Mac App Store distribusjon.
- App må signeres før den kan sendes til Apple.

DÉMO

ANNEIT

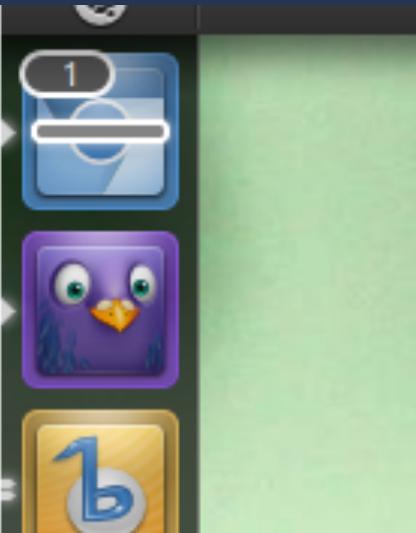
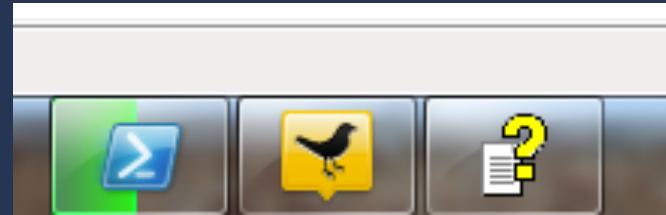
SMÅGODT

Windows Thumbnail Toolbar



```
var win = new BrowserWindow( { width: 800, height: 600 } );
win.setThumbarButtons( [ . . ] );
```

Progress Bar In Taskbar



```
var window = new BrowserWindow( { . . . } );  
window.setProgressBar(0.5);
```

Andre kule API-er

- **dialog** - native dialoger
- **global-shortcut** - globale keyboard shortcuts
- **power-monitor** - overvåke strømforbruk
- **power-save-blocker** - allways on
- **protocol** - egne protokoll handler i.e. minapp://foo
- **clipboard** - tilgang til clipboard
- **shell** - åpne mapper og filer

OPPSUMMERING

Lyst å lære mer?

- Getting Started guide på electron.atom.io
- Steg-for-steg tutorial som lager en Sound Machine App
- Signering av Mac Apps



Takk for meg!

github.com/follesoe/divemanager-edit

