



LECTURE 10

# IONIC + ELECTRON PUBLISH APP

DANIEL RYS  
JAN VÁCLAVÍK

# OVERVIEW

- Ionic + Electron
- Publish app

# ELECTRON

- For building **desktop apps**
- Supports **Linux, Mac OS X, Windows**
- Wrap **web app in chromium-based browser**
- Used by Slack, Avocode, Atom, Visual Studio Code, Postman,...

# ELECTRON SIZE PROBLEM

- **Ionic app:** ~10 MB
- **Electron app**
  - **Mac OS X:** ~120 MB
  - **Linux32:** ~60 MB
  - **Linux64:** ~55 MB
  - **Win ia32:** ~48 MB
  - **Win x64:** ~66 MB

# DOWNLOAD APP FROM LAST LECTURE

```
$ git clone https://github.com/jvaclavik/image-warholizer  
$ cd image-warholizer  
$ git checkout 2.0  
$ npm i
```

<https://github.com/jvaclavik/image-warholizer/releases/tag/2.0>

# INSTALL ELECTRON

```
$ npm install -g electron-prebuilt  
$ npm install -g electron-packager
```

# PREPARE

- Create **www/electron.js** for Electron configuration
- Create **www/package.json** for Node configuration

# ELECTRON CONFIGURATION

`www/electron.js`

```
var app = require('app');  
var BrowserWindow = require('browser-window');  
var mainWindow = null;  
app.on('window-all-closed', function () {  
    if (process.platform !== 'darwin') {  
        app.quit();  
    }  
});  
  
app.on('ready', function () {  
    mainWindow = new BrowserWindow({width: 1100, height: 900});  
    mainWindow.loadURL('file://' + __dirname + '/index.html');  
    mainWindow.on('closed', function () {  
        mainWindow = null;  
    });  
});
```



# UPDATE PACKAGE.JSON

`www/package.json`

```
{  
  "name": "Warholizer",  
  "version": "0.1.0",  
  "main": "electron.js"  
}
```

# UPDATE PACKAGE.JSON

www/package.json

```
{  
  "name": "Warholizer",  
  "version": "0.1.0",  
  "main": "electron.js"  
}
```

Which file should be executed



# RUN ELECTRON APP

```
$ cd www  
$ electron .
```

# ADD MENU

<http://electron.atom.io/docs/v0.28.0/api/menu/>

# ADD MENU

`www/menu.js`

```
function sayHello() { alert("Hello") }
const remote = require('electron').remote;
const Menu = remote.Menu;
menuTemplate = [
  {
    label: "Important menu",
    submenu: [
      {
        label: "Say hello",
        click: sayHello
      }
    ]
  }
];
if (process.platform === 'darwin')
  menuTemplate.unshift({
    label: "Basic",
    submenu: [
      {
        label: 'About ',
        role: 'about'
      }
    ]
  });
Menu.setApplicationMenu(Menu.buildFromTemplate(menuTemplate))
```

# ADD MENU SCRIPT

`www/index.html`

```
<script src="menu.js"></script>
```

# ADD ICON (FOR MAC OS X)

- convert **PNG to ICNS** (Apple icon format)
- <https://iconverticons.com/online/>

# ADD ICON (FOR MAC OS X)

- Use electron packager
- Copy icon to generated app

```
$ electron-packager . Warholizer --ignore=node_modules/  
electron-* --platform=darwin --arch=x64 --version=0.36.10  
--overwrite && cp ../resources/icon.icns Warholizer-  
darwin-x64/Warholizer.app/Contents/Resources/atom.icns
```



# ADD ICON (FOR MAC OS X)

- Use electron packager
- Copy icon to generated app

```
$ electron-packager . Warholizer --ignore  
electron-* --platform=darwin --arch=x64 --  
--overwrite && cp ../resources/icon.icns Warholizer.darwin-x64/Warholizer.app/Contents/Resources/atom.icns
```

Console hell :-)



# CREATE NPM SCRIPT

`www/package.json`

```
{
  "name": "Warholizer",
  "version": "0.1.0",
  "main": "electron.js",
  "scripts": {
    "start": "electron .",
    "build": "electron-packager . Warholizer --ignore=node_modules/electron-* --platform=darwin --arch=x64 --version=0.36.10 --overwrite && cp ../resources/icon.icns Warholizer-darwin-x64/Warholizer.app/Contents/Resources/atom.icns"
  },
  "devDependencies": {
    "asar": "^0.10.0",
    "electron-packager": "^3.2.0",
    "electron-prebuilt": "^0.25.2"
  }
}
```

# RUN NPM SCRIPT

```
$ npm run-script build
```

# WEBCAMERA IMPLEMENTATION

# WEBCAMERA IMPLEMENTATION

app/utility.js

```
export class Utility {  
  constructor() {};  
  static isElectron() {  
    return (window && window.process && window.process.type)  
      || !window.cordova;  
  };  
}
```

# WEBCAMERA IMPLEMENTATION

app/pages/choose-action/choose-action.html

```
<ion-navbar *navbar>
  <button menuToggle *ngIf="!isElectron()">
    <ion-icon name="menu"></ion-icon>
  </button>
  <ion-title>New Effect</ion-title>
</ion-navbar>

<ion-content padding class="choose-action">
  <button block (click)="getPicture('device')" *ngIf="!isElectron()">
    Choose from device
  </button>
  <button block (click)="getPicture('camera')">Take a picture</button>
  <video id="video" width="100%" autoplay *ngIf="isElectron()"></video>
  <button class="button button-block button-secondary"
id="snap" (click)="snapPhoto($event)" *ngIf="isElectron()">Snap Photo</button>
  <canvas class="hide" id="canvas" width="640" height="480"></canvas>
</ion-content>
```

# ADD UTILITY

app/pages/choose-action/choose-action.js

```
...
import {Utility} from '../../utility';

@Page({
  templateUrl: 'build/pages/warholize/warholize.html',
  providers: [GalleryProvider]
})
export class ChooseActionPage {
  ...
  isElectron() {
    return Utility.isElectron();
  }
}
```

# ADD UTILITY

app/pages/choose-action/choose-action.js

```
snapPhoto(event) {  
  var canvas = document.getElementById("canvas");  
  var context = canvas.getContext("2d");  
  context.drawImage(video, 0, 0, 640, 480);  
  this.nav.push(CropperPage, {imageData: canvas.toDataURL("image/jpeg")});  
}
```



# WEBCAMERA IMPLEMENTATION

app/pages/choose-action/choose-action.js

```
getPicture(source) {  
  ...  
  Camera.getPicture(options).then((imageData) => {  
    this.nav.push(CropperPage, {imageData: imageData});  
  }, (err) => {  
    if(source !== 'device'){  
      var video = document.getElementById("video");  
      var videoObj = { "video": true };  
      var videoErrorHandler = function(error) {  
        console.error("Failed to capture video: ", error.code);  
      };  
      if(navigator.getUserMedia) { // Standard  
        navigator.getUserMedia(videoObj, function(stream) {  
          video.src = stream;  
          video.play();  
        }, videoErrorHandler);  
      } else  
        if(navigator.webkitGetUserMedia) { // WebKit-prefixed  
          navigator.webkitGetUserMedia(videoObj, function(stream) {  
            video.src = window.webkitURL.createObjectURL(stream);  
            video.play();  
          }, videoErrorHandler);  
        }  
      } else {  
        ...  
      }  
    });  
  }  
}
```

# ADD UTILITY

app/pages/warholize/warholize.js

```
...
import {Utility} from '../../utility';

@Page({
  templateUrl: 'build/pages/warholize/warholize.html',
  providers: [GalleryProvider]
})
export class WarholizePage {
  ...
  isElectron() {
    return Utility.isElectron();
  }
}
```

# SAVE AS IMAGE

app/pages/warholize/warholize.js

```
applyEffect(orientation) {
  ...
  let wrh = new Warholizer({
    ...
    onRenderFinished: () => {
      loading.dismiss();
      if (Utility.isElectron()) {
        var canvases = document.querySelectorAll('canvas');
        for (var i = 0; i < canvases.length; i++) {
          var parent = canvases[i].parentElement;
          if (parent.id == 'previews') {
            var a = document.createElement('a');
            a.href = canvases[i].toDataURL('image/jpeg');
            a.download = 'stripes.jpg';
            a.appendChild(canvases[i]);
            parent.appendChild(a);
          }
        }
      }
    },
    cloneClickedHandler: (target) => {
      if (!Utility.isElectron()) this.showActionSheet(target);
    }
  });
  ...
}
```

# DISABLE SIDE MENU

app/app.js

```
...
import {Utility} from './utility';

@App({
  templateUrl: 'build/app.html',
  config: {}
})
class MyApp {
  static get parameters() {
    return [[IonicApp], [Platform], [MenuController]];
  }
  constructor(app, platform, menu) {
    this.menu = menu;
    ...
  }
  initializeApp() {
    this.platform.ready().then(() => {
      ...
      if (Utility.isElectron()) {
        this.menu.swipeEnable(false);
      }
    });
  }
  ...
}
```

# PUBLISH APPLICATION ON GOOGLE PLAY

# REGISTER GOOGLE PUBLISHER ACCOUNT

- Registration fee: \$25
- <http://developer.android.com/distribute/googleplay/start.html>

# PREPARE CONFIG.XML

- ID
- Name
- Version
- Icon & splash screen

# SIGNING APK



# GENERATE KEY

```
$ keytool -genkey -v -keystore warholizer.keystore -  
alias warholizer -keyalg RSA -keysize 2048 -validity  
10000
```

# GENERATE KEY

```
$ keytool -genkey -v -keystore warholizer.keystore -alias warholizer -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Jan Vaclavik
What is the name of your organizational unit?
  [Unknown]:  Userte.ch
What is the name of your organization?
  [Unknown]:  Userte.ch
What is the name of your City or Locality?
  [Unknown]:  Prague
What is the name of your State or Province?
  [Unknown]:  Czech Republic
What is the two-letter country code for this unit?
  [Unknown]:  CZ
Is CN=Jan Vaclavik, OU=Userte.ch, O=Userte.ch, L=Prague, ST=Czech Republic, C=CZ correct?

  [no]:  yes

Generating 2048 bit RSA key pair and self-signed certificate (SHA1withRSA) with a validity of 10000 days
    for: CN=Jan Vaclavik, OU=Userte.ch, O=Userte.ch, L=Prague, ST=Czech Republic, C=CZ
Enter key password for <warholizer>
    (RETURN if same as keystore password):
Re-enter new password:
[Storing warholizer.keystore]
```

# RELEASE BUILD

```
$ cordova build android --release
```

# APK SIGNING

```
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg  
SHA1 -keystore warholizer.keystore android-release-  
unsigned.apk warholizer
```

# SIGNING CHECK

```
$ jarsigner -verify -verbose -certs android-release-unsigned.apk
```

# CREATE APK

```
$ zipalign -v 4 android-release-unsigned.apk android-release-signed.apk
```

# WHAT NEEDS TO FILL IN GOOGLE PLAY

- **Name**
- **Short description** ( $\leq 80$  characters)
- **Full description** ( $\leq 4000$  characters)
- **Screenshots** (2–8 images)
  - On the phone: Hold power and volume down button
- **Icon** ( $512 \times 512\text{px}$ )
- **Main graphics** ( $1024 \times 500\text{px}$ )
- **Application type** (Application/game)
- **Category**
- **Content rating**
- **E-mail**

# DEPLOY ON GOOGLE PLAY

<https://play.google.com/apps/publish/>



# DEPLOY ON APP STORE

The screenshot shows the Google Play Developer Console interface for the app 'Bubbleology'. The top header includes the Google Play logo, the text 'Google Play Developer Console', a search bar, and notification, help, and profile icons. The left sidebar contains navigation links: 'Statistiky', 'Akvizice uživatelů', 'Hodnocení a recenze', 'Selhání a chyby ANR', 'Tipy k optimalizaci', 'Cloud Test Lab', 'Promo akce', 'APK', 'Záznam v obchodu' (highlighted), 'Experimenty', 'Hodnocení obsahu', 'Cena a distribuce', 'Produkty v aplikacích', and 'Služby a rozhraní API'. The main content area is titled 'ZÁZNAM V OBCHODU' and 'PODROBNOSTI PRODUKTU'. It shows the app's name 'Bubbleology', its package name 'com.usertechnologies.bubbleology2', and its status 'PUBLIKOVÁNO' with a date '11. března 2016'. There are buttons for 'Zobrazit v obchodu Play' and 'Zrušit publikování aplikace'. A 'Standardní publikování' dropdown and an 'Odeslat aktualizaci' button are also present. The 'PODROBNOSTI PRODUKTU' section includes a language selector set to 'Čeština - cs-CZ' and a 'Spravovat překlady' button. It displays the app's name, short description, and full description in Czech. The full description includes a loyalty program announcement and a call to action to register. At the bottom, there are links to 'Přečtěte si zásady ohledně metadat' and 'Pokud je aplikace nebo záznam v obchodu vhodný k předběžnému oznámení týmu kontroly aplikací na Google Play'.

Google Play Developer Console

Bubbleology  
com.usertechnologies.bubbleology2 [Zobrazit v obchodu Play](#)  
PUBLIKOVÁNO 11. března 2016 [Zrušit publikování aplikace](#)

Standardní publikování ▼  
[Odeslat aktualizaci](#)

Statistiky  
Akvizice uživatelů  
Hodnocení a recenze  
Selhání a chyby ANR  
Tipy k optimalizaci  
Cloud Test Lab  
Promo akce

APK

**Záznam v obchodu**  
Experimenty  
Hodnocení obsahu  
Cena a distribuce  
Produkty v aplikacích  
Služby a rozhraní API

**ZÁZNAM V OBCHODU**

PODROBNOSTI PRODUKTU

Pole označená \* musí být před zveřejněním vyplněna.

Čeština - cs-CZ Jazyky (2) ▼ [Spravovat překlady ▼](#)

**Název\***  
Čeština - cs-CZ Bubbleology  
11 z 30 znaků

**Krátký popis\***  
Čeština - cs-CZ Bubbleology vám přináší novou mobilní aplikaci pro váš chytrý telefon.  
71 z 80 znaků

**Úplný popis \***  
Čeština - cs-CZ

S aplikací máte svůj věrnostní program stále při ruce. Registrujte se a využívejte výhod věrnostního programu přímo z vašeho telefonu. Na pobočce stačí jen ukázat unikátní QR kód. S aplikací Bubbleology si k nám najdeš cestu ať už jsi kdekoliv. Nezapomeň si ale zapnout GPS! Seznámit se můžete s naší kompletní nabídkou a vytvořit si jedinečný mix chutí. Nebo dejte na doporučení našich čajových odborníků a vyberte jednu z deseti doporučovaných kombinací. Po ochutnání nezapomeňte drink ohodnotit!

**PROČ SE REGISTRovat DO VĚRNOSTNÍHO PROGRAMU BUBBLEOLOGY?**  
Získáš kredit v hodnotě 10 % z každého nákupu zpět! S aplikací navíc nebudeš potřebovat plastovou kartu.

**Funkce aplikace:**  
• Věrnostní program přímo v telefonu

1846 z 4000 znaků

Přečtěte si [zásady ohledně metadat](#), abyste předešli jejich porušení. Před odesláním aplikací si také projděte všechny ostatní [programové zásady](#).

Pokud je aplikace nebo záznam v obchodu [vhodný k předběžnému oznámení](#) týmu kontroly aplikací na Google Play, před publikováním nás [kontaktujte](#).

PUBLISH ON APP STORE

# REGISTER APPLE DEVELOPER ACCOUNT

- Register developer account
- \$99 per year
- <https://developer.apple.com/support/compare-memberships/>

# PUBLISH ON APP STORE

- Create **production certificate** of your company
- Create **identifier** for the app
- Create **distribution provisioning profile**
- Deal with **xCode**
- Upload to **iTunesConnect**
- **Publish on App Store** (takes a week)

# PUBLISH ON APP STORE

- **iTunesConnect**
  - For publishing apps
  - [itunesconnect.apple.com](https://itunesconnect.apple.com)
- **Developer Account**
  - For generating certificates
  - [developer.apple.com](https://developer.apple.com)

QUESTIONS?



JAN VÁCLAVÍK

@janvaclavik

DANIEL RYS

@danielrys

WWW.USERTECHNOLOGIES.COM