

HAMPUS NILSSON

# GETTING STARTED WITH ANGULAR 2 FOR BACKEND DEVS

---

<http://hjnilsson.com/downloads/2017-12-15-angular.pdf>

## DISPOSITION

- ▶ Tools
- ▶ The necessary evils: npm & webpack
- ▶ Angular 2+
- ▶ Fun ~ !

## MODERN FRONTEND

- ▶ Very different from the jQuery days
- ▶ Do you know HTML/CSS?
- ▶ In general, you use classes, build views & controllers in a very analogous way to native UI frameworks
- ▶ To support this change from ye olde' days, tooling is necessary

# DEVELOPMENT TOOLS

TEXT

---

# DEVELOPMENT TOOLS

OBVIOUSLY THERE ARE OPTIONS

## DEVELOPMENT TOOLS

- ▶ Chrome
  - ▶ Best debugger / inspector
- ▶ Visual Code
  - ▶ Not obviously the best, but tightly integrated and updated often
  - ▶ Microsoft made

# NODE / NPM

The maven of Frontend

## NPM / WEB PACK

- ▶ Node is JavaScript run as a program on your local machine, not in the browser
- ▶ Node is what powers JavaScript based servers, but also comes with MANY packages for other purposes

*“The App Store of the command line”*

- ▶ The most important function of node for frontend development is compilation & package management



## GO GET NODE!

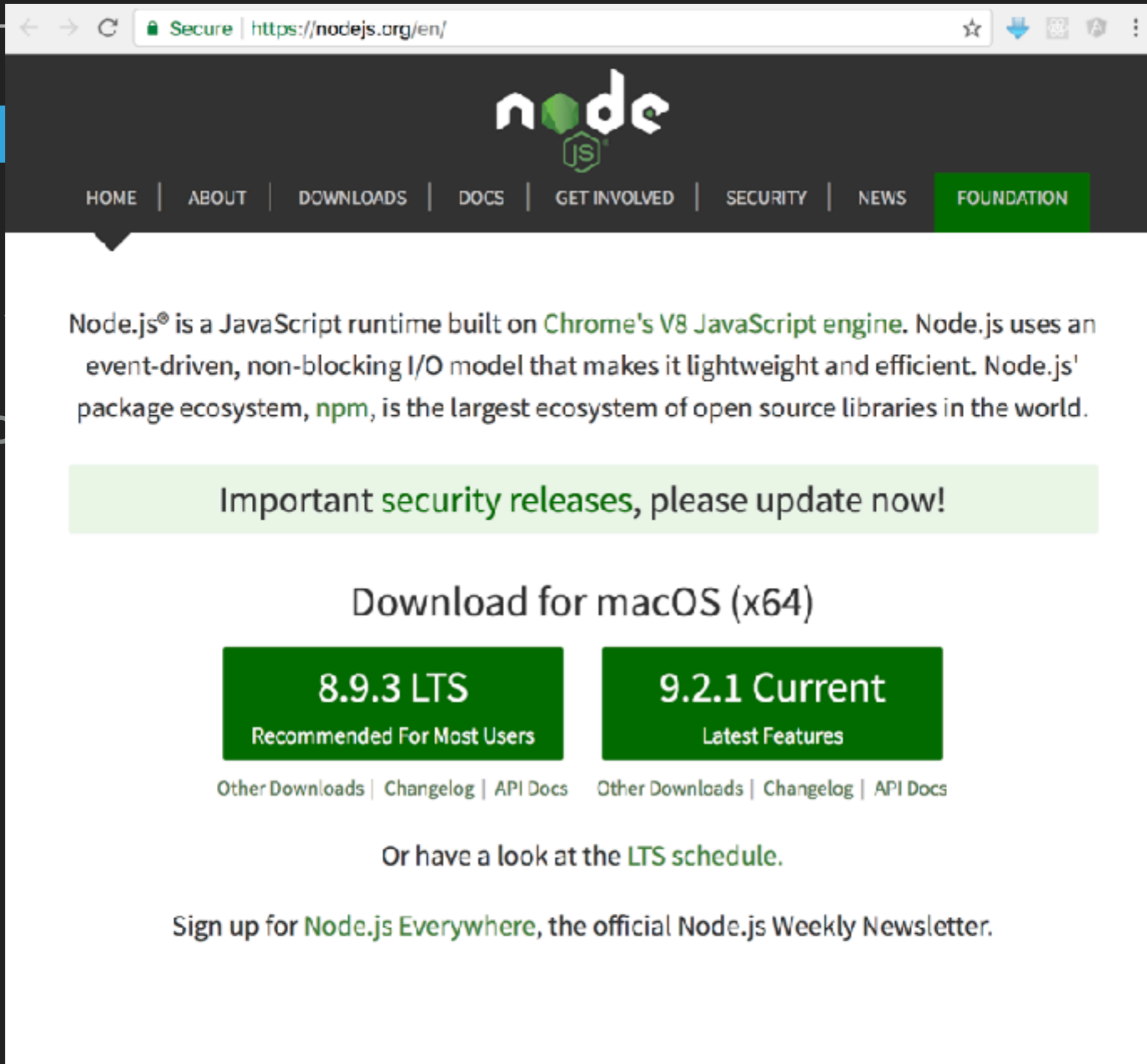
- ▶ Make sure you got an updated version
- ▶ Mystic errors if you use old versions.

TEXT

GO GET N

▶ Make

▶ Mystic



The image shows a screenshot of the Node.js website homepage. The browser's address bar displays "Secure | https://nodejs.org/en/". The website's navigation menu includes links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The main content area features a description of Node.js as a JavaScript runtime built on Chrome's V8 JavaScript engine, highlighting its event-driven, non-blocking I/O model and the npm package ecosystem. A prominent green banner alerts users to "Important security releases, please update now!". Below this, the "Download for macOS (x64)" section offers two options: "8.9.3 LTS Recommended For Most Users" and "9.2.1 Current Latest Features". Each option includes links for "Other Downloads", "Changelog", and "API Docs". At the bottom, there is a link to "Sign up for Node.js Everywhere, the official Node.js Weekly Newsletter."

Secure | https://nodejs.org/en/

node JS

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on **Chrome's V8 JavaScript engine**. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

Important **security releases**, please update now!

Download for macOS (x64)

**8.9.3 LTS**  
Recommended For Most Users

**9.2.1 Current**  
Latest Features

Other Downloads | Changelog | API Docs    Other Downloads | Changelog | API Docs

Or have a look at the **LTS schedule**.

Sign up for **Node.js Everywhere**, the official Node.js Weekly Newsletter.

## NODE COMMANDS

- ▶ `npm install -g <package>`
- ▶ `npm install --save <package>`
  
- ▶ ... the rest you'll find on Stack Overflow

## NODE COMMANDS

▶ `npm install -g <package>`

GLOBAL INSTALL

▶ `npm install --save <package>`

▶ ... the rest you'll find on Stack Overflow

## NODE COMMANDS

▶ `npm install -g <package>`

GLOBAL INSTALL

▶ `npm install --save <package>`

LOCAL INSTALL

▶ ... the rest you'll find on Stack Overflow

## NODE COMMANDS

▶ `npm install -g <package>`

GLOBAL INSTALL

▶ `npm install --save <package>`

LOCAL INSTALL

Word of caution: `npm update` doesn't do what you think it does  
(It does not upgrade)

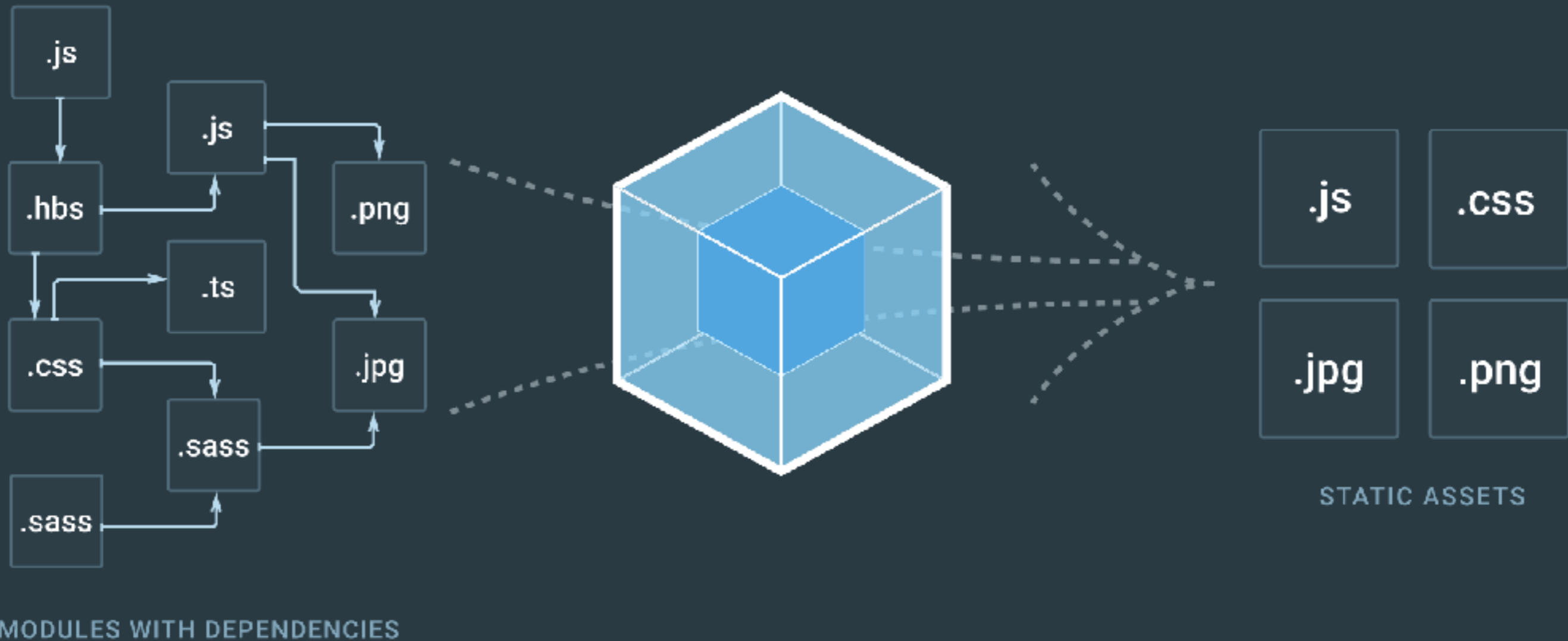
▶ ... the rest you'll find on Stack Overflow

**WEBPACK**

## WEBPACK

- ▶ The code you write is not able to run in the browser
- ▶ Just like Java / C# etc. can not run directly as an executable, you need to compile it!
- ▶ Webpack is the tool that compiles your code for the browser





- ▶ Webpack handles all frontend resources, not just code

# YOU DON'T SEE THIS ANY MORE

```
<script src="/resources/external/libs/lodash/lodash.4.6.1.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/fra_cdn/jquery-2.2.2.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/menu-aim/menu-aim.2013-modified.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/ladda/1.0.0/spin.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/ladda/1.0.0/ladda.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/moment/2.5.1/moment.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/moment/2.5.1/nb.js" type="text/javascript"></script>
<script src="/resources/external/libs/chartjs/Chart.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/daterangepicker/2013-10-12/daterangepicker.js" type="text/javascript"></script>
<script src="/resources/external/libs/angular/1.6.6/angular.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/angular/1.6.6/angular-route.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/angular/1.6.6/angular-sanitize.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/angular/1.6.6/angular-animate.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/bootstrap/3.3.7/js/bootstrap.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/bootstrap/datepicker/1.7.1/js/bootstrap-datepicker.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/bootstrap/datepicker/1.7.1/locales/bootstrap-datepicker.nb.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/bootstrap/bsui 2.5.0/ui-bootstrap-custom-tpls-2.5.0.js" type="text/javascript"></script>
<script src="/resources/external/libs/jq_fileupload/9.8.1/vendor/jquery.ui.widget.js" type="text/javascript"></script>
<script src="/resources/external/libs/jq_fileupload/9.8.1/jquery.fileupload.js" type="text/javascript"></script>
<script src="/resources/external/libs/jq_fileupload/9.8.1/jquery.iframe-transport.js" type="text/javascript"></script>
<script src="/resources/external/libs/nprogress/0.1.2/nprogress.js" type="text/javascript"></script>
<script src="/resources/external/libs/big/big-2.5.0.js" type="text/javascript"></script>
<script src="/resources/external/libs/clipboard/1.5.5/clipboard.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/commonmark/0.22.0/commonmark.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/mousetrap/1.5.3/mousetrap.min.js" type="text/javascript"></script>
<script src="/resources/external/libs/mousetrap/1.5.3/plugins/global-bind/mousetrap-global-bind.min.js" type="text/javascript"></script>
```

TEXT

---

YOU SEE THIS:

```
.....<link href='https://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>  
.....<link rel="stylesheet" type="text/css" href="/Static/dist/buf15.css?v=1.0.0.25362" />  
.....<script type="text/javascript" src="/Static/dist/plain-js-libs.js?v=1.0.0.25362" defer></script>
```

## NG-CLI

- ▶ Using web pack directly is *\*very\** complicated
- ▶ Angular comes with a tool to abstract webpack away
- ▶ `npm install -g @angular/cli`

THIS IS A GLOBAL TOOL, SO WE USE THE -G OPTION

## NG-CLI

- ▶ Getting started
- ▶ `ng new NAME`
- ▶ `ng serve`

---

**LET'S DO IT LIVE!**

**Developer hubris**

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---



# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

**Module visibility**

Should this be available outside file? Then export

**Interfaces are done with decorators**

Pretty much same as C# properties

with constructor arguments

**Import dependencies**

You (almost) always specify exactly what you need

**Specify where to find resources**

(Angular specific ...)

You need

ictionaries, strings etc are all exactly like JSON

(Angular specific ...)

**Class visibility**

Access inside a class is same as C# / Java (public is default)

Types are specified with :

```
}
export class AppComponent {
  member
```

```
e';
p';
```

```
this
}
```

```
public
this.ye
}
```

```
private y
this.me
}
```

```
}
```

```
tyInject: OfType
pClient) {
another value';
```



# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

**Lists, dictionaries, strings  
etc are all exactly like  
JSON**

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

```
export class AppComponent implements OnInit {
  memberProperty = 'aValue';
```

You must always use this

```
    memberList = ['hello', 'great'];
  constructor(private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

Types are specified  
with :

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
```

```
export class AppComponent {
  memberProperty = 'aValue';
```

```
  memberList = ['Computers'];
```

```
  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }
```

```
  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }
```

```
  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
```

**Dependency injection is done  
with constructor arguments**

(Angular specific ...)

```
}
```



# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

```
export class AppComponent implements OnInit {
  memberProperty = 'aValue';
```

```
  memberList = ['Computas', 'is', 'great'];
```

```
  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }
```

```
  public
    this.ye
  }
```

```
  private y
    this.me
  }
```

```
}
```

## Class visibility

Access inside a class is same  
as C# / Java (public is default)

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

## Module visibility

Should this be available outside this file? Then export

```
OnInit } from '@angular/core';  
from '@angular/common/http';  
  
,  
'component.html',  
omponent.scss']
```

```
})  
export class AppComponent implements OnInit {  
  memberProperty = 'aValue';  
  
  memberList = ['Computas', 'is', 'great'];  
  
  constructor(private dependencyInject: OfType,  
              private http: HttpClient) {  
    this.memberProperty = 'to another value';  
  }  
  
  public ngOnInit() {  
    this.yesYouNeedThis(this.memberList);  
  }  
  
  private yesYouNeedThis(aList : String[]) {  
    this.memberProperty = aList.join(' ');  
  }  
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

Interfaces are done  
with `implements`

(You can inherit, but  
uncommon)

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

Import dependencies

You (almost) always  
specify exactly what  
you need



# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

Class decorator

Pretty much same as  
C# properties

```
import { Component, NgModule } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

```
export class AppComponent implements OnInit {
```

```
  member
```

**Specify where to find  
resources**

```
  member
```

```
    's', 'great'];
```

```
  const
```

**(Angular specific ...)**

```
    @Inject( OfType,
```

```
    HttpClient) {
```

```
    this
```

```
      'another value';
```

```
  public ngOnInit() {
```

```
    this.yesYouNeedThis(this.memberList);
```

```
  }
```

```
  private yesYouNeedThis(aList : String[]) {
```

```
    this.memberProperty = aList.join(' ');
```

```
  }
```

```
}
```

# ANGULAR 2 AND TYPESCRIPT IN ONE SLIDE

---

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  memberProperty = 'aValue';

  memberList = ['Computas', 'is', 'great'];

  constructor(private dependencyInject: OfType,
               private http: HttpClient) {
    this.memberProperty = 'to another value';
  }

  public ngOnInit() {
    this.yesYouNeedThis(this.memberList);
  }

  private yesYouNeedThis(aList : String[]) {
    this.memberProperty = aList.join(' ');
  }
}
```

---

**LET'S DO IT LIVE!**

**... AND SOMETHING FUN THIS TIME**

**Developer hubris**

TEXT

# COMPUTAS SPONSORED GITHUB SEARCH!



Github repos:

computas

Go!



[bjartwolf/computas-play](#), 2 stars



[espenmeidell/IOT\\_Hackathon](#), 0 stars



[bflugon/iot-hackathon](#), 0 stars

# THE FINAL TEMPLATE

```
<!--The content below is only a placeholder and can be replaced.-->
<div class="container">
  <div style="text-align:center">
    <h1>
      Hello
    </h1>
    <svg class="logo" focusable="false" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 389.33334 121.33333">
      <path d="M22.255 66.15h13.657s-4.647 8-9.505 8.23c-10 .568-14.522 1.292-15.814 9.443-1.65 5.718 1 7.505 10.522 7.937 7.926.39 6.697 8.24 6.697 8.24h-14.22c-11 0-14.306-6" data-bbox="22.255 66.15 136.862 121.33333"/>
      <path d="M303.724 13.76c-2.37 5.864-8.047 9.713-14.37 9.75-14.562.22-21 2.792-23.818 10.922-1.572 4.51-.54 8.328 2.12 9.37 4.016 1.557 13.156-10.214 20.156-3.65 7 5.556-6" data-bbox="303.724 13.76 389.33333 121.33333"/>
    </svg>
  </div>
  <div class="row">
    <div class="col-md-6">
      <h2>Github repos:</h2>
    </div>
    <div class="col-md-6">
      <form class="input-group">
        <input type="text" name="searchFor" class="form-control" placeholder="Search for..." aria-label="Search for..." [(ngModel)]="searchString">
        <span class="input-group-btn">
          <button class="btn btn-secondary" type="submit" (click)="onSearch()">Go!</button>
        </span>
      </form>
    </div>
    <div class="col-12">
      <ul class="repo-list list-group">
        <li *ngFor="let repo of repositories" class="list-group-item">
          
          <a href="{{repo.html_url}}">{{repo.full_name}}</a>
          <span>, {{repo.stargazers_count}} stars</span>
        </li>
      </ul>
    </div>
  </div>
</div>
</div>
```



# THE STYLING

## ▶ styles.scss (global)

```
/* You can add global styles to this file, and also import other style files */  
@import '../node_modules/bootstrap/scss/bootstrap.scss';
```

## ▶ app.component.scss (local to component)

```
$logo-width: 300px;  
  
.logo {  
  width: $logo-width;  
}  
  
img {  
  width: 100px;  
}
```

# THE APP MODULE

- ▶ Need to provide HttpClient and FormsModule for extra functionality.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

# PACKAGE.JSON

```
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^5.0.0",
    "@angular/common": "^5.0.0",
    "@angular/compiler": "^5.0.0",
    "@angular/core": "^5.0.0",
    "@angular/forms": "^5.0.0",
    "@angular/http": "^5.0.0",
    "@angular/platform-browser": "^5.0.0",
    "@angular/platform-browser-dynamic": "^5.0.0",
    "@angular/router": "^5.0.0",
    "bootstrap": "^4.0.0-beta.2",
    "core-js": "^2.4.1",
    "rxjs": "^5.5.2",
    "zone.js": "^0.8.14"
  },
  "devDependencies": {
    "@angular/cli": "1.6.1",
    "@angular/compiler-cli": "^5.0.0",
    "@angular/language-service": "^5.0.0",
    "@types/jasmine": "~2.5.53",
    "@types/jasminewd2": "~2.0.2",
```

**WE INSTALLED  
BOOTSTRAP**

# TASKS.JSON

- ▶ Make your life easier

```
1  {
2  ...// See https://go.microsoft.com/fwlink/?LinkId=733558
3  ...// for the documentation about the tasks.json format
4  ..."version": "2.0.0",
5  ..."tasks": [
6  ...  {
7  ...    "type": "npm",
8  ...    "script": "start",
9  ...    "problemMatcher": []
10 ...  }
11 ... ]
12 }
```

**COMPARISONS**

## Angular

## React

## Vue

### Strengths

- Batteries included
- Strong TypeScript support
- Strong backing by Google
- One way to do things

- Simplest
- Very widely used
- Battle tested
- Strong backing by Facebook, Pinterest, LinkedIn

- Simple
- Very quick to get started
- More batteries than React (less than Angular)

### Weaknesses

- Very complex, many moving parts
- Opinionated

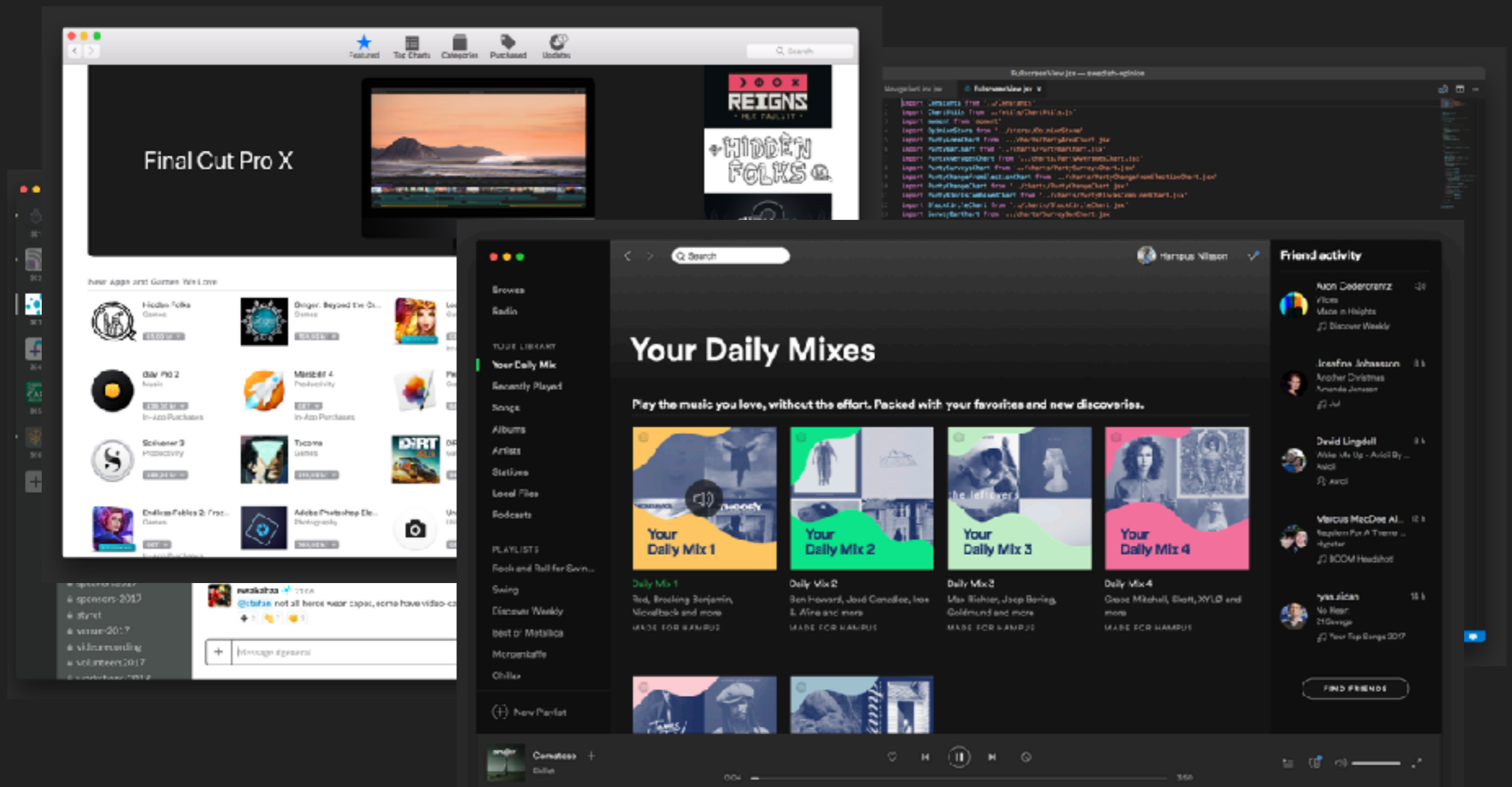
- Needs supplementary libraries
- Complex setup

- No huge backer
- Tooling is slightly behind

All are fast!

# ELECTRON

- ▶ Many apps nowadays are actually React apps



## ELECTRON

- ▶ Many benefits, easy styling, great tools, easy to update
- ▶ Works in both desktop and browser
- ▶ Potential for clients?



## TAKEAWAYS

- ▶ Install node
- ▶ Get ng cli, `ng new github-searcher`
- ▶ Take the tour of heroes
- ▶ TypeScript is very similar to Java/C# (and for regular JavaScript, don't type the types!)
- ▶ Well designed APIs make querying easy
- ▶ You learn the errors... after a long time.

---

## RESOURCES (JUST GOOGLE THESE THINGS)

LIKE, WHO REMEMBERS LINKS?

- ▶ Angular Tour of Heroes
- ▶ Visual code
- ▶ ng-cli
- ▶ TypeScript
- ▶ Protractor / Webdriver IO
- ▶ Vue / React / Electron

# FIN

Hampus Nilsson

<https://hjnilsson.com/downloads/2017-12-15-angular.pdf>